

Usable Access Control

Vom Fachbereich Informatik
der Technischen Universität Darmstadt genehmigte

DISSERTATION

zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.)

von

Dipl.-Inform. Matthias Beckerle

geboren in Bensheim, Deutschland



Referenten

Prof. Dr. Max Mühlhäuser (Technische Universität Darmstadt)
Prof. Dr. Lujo Bauer (Carnegie Mellon University)

Tag der Einreichung: 29.10.2013
Tag der mündlichen Prüfung: 19.12.2013

Darmstadt 2014
Hochschulkennziffer D17

Abstract

The research described in this work can significantly simplify and facilitate the creation and configuration of secure access control rule sets.

Access control is used to provide confidential data or information only to authorized entities and deny access otherwise. Access control mechanisms can be configured with access control rule sets that need to be created and maintained by the users or administrators.

The research commences by answering the first research question:

1. How can access control be integrated into future products?

Basic concepts are presented and integrated into a holistic design. The latter is embedded into a general framework, which was developed by an academia-industry consortium, and in which the author participated.

Questions arise regarding usability aspects of access control mechanisms. An analysis of security services in the beginning of this dissertation shows that, especially for access control mechanisms that are managed by casual users, a high level of usability is required because individual preferences of the data owner have to be taken into account.

Analysis of how the core security objectives (see Section 2.2) can be achieved identifies a usability gap regarding the generation and configuration of access control rule sets. Automation is not fully possible because individual preferences of users need to be considered.

Related research questions are:

2. What are the requirements for usable access control rule sets?

3. What are formally founded quantifiable measurements for those requirements, and how can these measurements be used to support users in generating of usable access control rule sets?

To answer these questions, a systematic analysis of expert opinions and related work was performed. The results of that analysis were grouped into categories and further refined into six informal requirements. The six informal requirements were mathematically formalized and six associated sets with respective linear metrics were derived. These formal tools are used to automatically calculate additional information about the actual access control rule set to support users in generating and optimizing the rule set properly. Two user studies were carried out to validate and evaluate the research and the findings presented in this work. They demonstrate that our metrics help users generate statistically significant better rule sets.

The dissertation concludes with an outlook and a vision for further research in usable access control rule set configuration.

Zusammenfassung

Die in dieser Dissertation dargestellte Forschung erleichtert wesentlich die Erstellung und Konfiguration von sicheren Zugriffskontrollregeln. Zugriffskontrolle wird benötigt um sicherzustellen, dass nur autorisierte Entitäten auf vertrauliche Daten oder Informationen zugreifen können. Zugriffskontrollmechanismen können mit Hilfe von Zugriffskontrollregelsätzen konfiguriert werden, welche von Administratoren oder Benutzern erstellt und gewartet werden müssen.

Diese Forschungsarbeit beginnt mit dem Beantworten der einleitenden Forschungsfrage:

1. Wie kann Zugriffskontrolle in zukünftige Produkte integriert werden?

Hierfür werden grundlegende Konzepte vorgestellt und in ein ganzheitliches Design integriert. Im Anschluss sind diese Konzepte in ein Programmiergerüst integriert worden, welches von einem Team aus akademischen und industriellen Partnern erstellt wurde, zu welchem auch der Autor gehörte.

Fragestellungen bezüglich der Benutzbarkeit von Zugriffskontrollmechanismen werden erörtert. Eine Analyse von Sicherheitsdiensten zu Beginn dieser Dissertation zeigt, dass insbesondere Zugriffskontrollmechanismen, welche von nicht professionellen Nutzern betreut werden, ein hohes Maß an Benutzbarkeit benötigen, da individuelle Präferenzen der Datenbesitzer berücksichtigt werden müssen.

Analysen, welche sich damit befassen, wie grundlegende Sicherheitsziele erreicht werden können, zeigen, dass die Erstellung und Konfiguration von Zugriffskontrollregelsätzen schwierig sein kann, da eine

vollständige Automatisierung hier nicht möglich ist, da die individuellen Präferenzen der Benutzer berücksichtigt werden müssen.

Die zugehörigen Forschungsfragen lauten:

2. Was sind die Anforderungen an benutzbare Zugriffskontrollregelsätze?
3. Was sind formal fundierte und quantifizierbare Messwerte für solche Anforderungen, und wie können diese Messwerte Benutzern helfen, benutzbare Zugriffskontrollregelsätze zu generieren?

Um diese Fragen zu beantworten wurde eine systematische Analyse sowohl von Expertenmeinungen als auch von verwandten Arbeiten durchgeführt. Die Resultate dieser Analyse wurden kategorisiert und zu sechs informellen Anforderungen für sichere und benutzbare Zugriffskontrollregelsätze weiterentwickelt. Diese sechs informellen Anforderungen werden mathematisch formalisiert und zu sechs linearen Metriken konsolidiert. Diese formellen Werkzeuge werden genutzt, um automatisiert Zusatzinformationen zu berechnen, welche Benutzer darin unterstützen können, Regelsätze zu erstellen und zu konfigurieren. Zwei Benutzerstudien wurden zur Validierung und Evaluierung der Forschung und der Resultate dieser Arbeit durchgeführt. Sie zeigen, dass die hier vorgestellten formellen Werkzeuge Benutzer darin unterstützen, signifikant bessere Regelsätze zu generieren.

Diese Dissertation schließt mit einem Ausblick und einer Vision für zukünftige Forschung im Bereich der benutzbaren Zugriffskontrollregelsatzkonfiguration.

Contents

Abstract	III
Zusammenfassung	VI
Contents	VII
List of Figures	XI
List of Tables	XIII
List of Abbreviations	XV
Acknowledgements	XVII
Remarks and List of Publications	XXI
1. Introduction	25
1.1. Motivation	27
1.2. Research Questions	29
1.3. Scientific Contributions	29
1.4. Research Method	29
1.5. Thesis Structure	31
2. Background	33
2.1. Pervasive Computing	33
2.1.1. Smart Products	34
2.1.2. Security for Pervasive Computing	36
2.2. Core Security Objectives	36
2.2.1. Confidentiality	36
2.2.2. Integrity	37
2.2.3. Authenticity	37
2.2.4. Authorization	37
2.3. Achieving Security Objectives for Smart Products	38
2.3.1. Approach for Confidentiality	38

2.3.2.	Approach for Integrity	39
2.3.3.	Approach for Authenticity	39
2.3.4.	Approach for Authorization	39
2.3.5.	Conclusion and Analysis	40
2.4.	Access Control Models	40
2.4.1.	Blacklists	41
2.4.2.	MAC/DAC	41
2.4.3.	RBAC	42
2.4.4.	ABAC	42
2.4.5.	Hybrid Approaches	43
2.4.6.	Access Control Model Conclusion	43
2.5.	Rule Learning	43
2.5.1.	Machine Learning	44
2.5.2.	Rule Learner	45
2.6.	State of the Art for Usable Access Control	46
2.7.	Summary	48
3.	Access Control for Smart Products	51
3.1.	Challenges and Goals	52
3.1.1.	Security Aspects of Smart Products	53
3.1.2.	Usability Aspects of Smart Products	55
3.2.	Concepts for SmartProducts Security	56
3.2.1.	Layered and Cooperative Security	57
3.2.2.	Usable Access Control	60
3.3.	SmartProducts Security Design	60
3.3.1.	SmartProducts Access Handling	62
3.3.2.	Security Administration	64
3.4.	SmartProducts Desing Evaluation	67
3.4.1.	Proof-of-Concept Prototype	68
3.4.2.	Functional Test Cases & Results	71
3.5.	Summary	71

4. Informal Requirements for Usable AC Rule Sets	73
4.1. Pilot Study	73
4.1.1. Methodology	74
4.1.2. Results	74
4.2. Definition of Goals	76
4.2.1. (G1) Allow no more than the owner wants to be allowed.	76
4.2.2. (G2) Allow everything the owner wants to be allowed.	77
4.2.3. (G3) Make sure that a rule is not fully covered by another rule of the same rule set.	77
4.2.4. (G4) Two rules within the same rule set must not conflict.	77
4.2.5. (G5) Minimize the number of rule set elements.	78
4.2.6. (G6) Minimize maintenance effort in a changing system.	78
4.3. On Goals and Derived Metrics	79
4.4. Summary	80
5. Quantifiable Metrics and Sets for Usable AC Rule Sets	81
5.1. Basic Building Blocks	82
5.2. Derived Building Blocks	83
5.3. Access Decision Sets	85
5.4. Security and Usability Metrics	86
5.5. The Cost of Wrong Access Decisions	88
5.6. Summary	89
6. Evaluation of Formal Metrics and Sets	91
6.1. Example	92
6.2. Validation of the Formal AC Rule Set Definitions	97
6.2.1. User Study 1 - Optimizing Rule Sets	97
6.2.2. User Study 2 - Approach Versus Experts	100
6.2.3. Results and Evaluation	101
6.3. Remarks	104

6.4. Discussion	105
6.5. Summary	107
7. Outlook and Conclusion	109
7.1. Data Accumulation / Getting Owners' Intention	109
7.2. Rule Extraction and Optimization	110
7.3. Rule Set Inspection	111
7.4. Future Vision: Interactive Rule Learning	111
7.4.1. Example	112
7.5. Conclusion	116
A. Appendix	119
A.1. User Study Rule Sets	119
Bibliography	129
Affirmation / Ehrenwörtliche Erklärung	137
Wissenschaftlicher Werdegang des Verfassers	139

List of Figures

2.1.1.SmartProducts Overview [SmartProducts]	35
3.2.1.Overview of Relationships between Goals, Concepts, and Design	58
3.2.2.Vertical and Horizontal Security Layers	59
3.3.1.Access Manager Architecture.	61
3.3.2.Access Handling Component Overview.	63
3.3.3.Access Handling Use Case.	65
3.3.4.Security Administration Module.	66
3.4.1.Cocktail Companion Overview	68
3.4.2.Access Manager: Proof-of-Concept	69
6.1.1.A graphical representation of the file system. It shows the files an entity should have access to (below the file name) and the non-default values $cost_{SGI}$ (in red). The default value is 10.	95
6.2.1.Box plot showing the results of User Study 1. They are presented with 0.95 confidence interval. The WS group (with support of our metrics and sets) performed sig- nificantly better (lower values) regarding the cost score than the WOS group (without support of our metrics and sets).	103

List of Tables

1.1. Methodology	30
3.1. Basic Goals related to Concepts	57
3.2. Access Control Functional Test Cases	70
5.1. Security and Usability Metrics: S_{Gi}	89
6.1. Entity–Attribute–Relationship Table. The ‘x’ marks indicate that a given attribute (column) is associated with a given entity (row), e.g., entity 1 has attributes $A3$, $A4$ and $A7$	93
6.2. A description of the file system showing which files an entity should have access to and the $cost_{S_{G1}}$, if an unauthorized access is allowed, assigned to each file. The $cost_{S_{G2}}$, if an authorized access is denied, assigned to each file, is 5 for each file.	94
6.3. Access Control Rule Set One	94
6.4. Usability scores computed using the metrics in Chapter 5 of Access Control Rule Set One	95
6.5. Access Control Rule Set Two	96
6.6. Usability scores computed using the metrics in Chapter 5 of Access Control Rule Set Two	96

6.7. Spearman's rank correlation test between the automatically produced rankings and the rankings obtained in User Study 2. <i>Proposal</i> refers to the automatically produced ranking and <i>Result 1</i> to <i>Result 4</i> to the results obtained from IT support professionals. $N = 18$ for all cases. A correlation coefficient of 1.000 would represent perfect correlation to the proposal. A correlation coefficient of 0.000 would represent no correlation at all.	100
6.8. Independent Samples t-test	102
7.1. Interactive Rule Learning Rule Set 1	114
7.2. Interactive Rule Learning Rule Set 2	115
7.3. Interactive Rule Learning Rule Set 3	115
7.4. Interactive Rule Learning Rule Set 4	115
A.1. User Study - Rule Set 1	119
A.2. User Study - Rule Set 2	120
A.3. User Study - Rule Set 3	120
A.4. User Study - Rule Set 4	120
A.5. User Study - Rule Set 5	121
A.6. User Study - Rule Set 6	121
A.7. User Study - Rule Set 7	122
A.8. User Study - Rule Set 8	122
A.9. User Study - Rule Set 9	123
A.10. User Study - Rule Set 10	124
A.11. User Study - Rule Set 11	125
A.12. User Study - Rule Set 12	125
A.13. User Study - Rule Set 13	126
A.14. User Study - Rule Set 14	126
A.15. User Study - Rule Set 15	127
A.16. User Study - Rule Set 16	127
A.17. User Study - Rule Set 17	127
A.18. User Study - Rule Set 18	128

List of Abbreviations

ABAC	Attribute Based Access Control
AC	Access Control
AH	Access Handler
BH	Blacklist Handler
DAC	Discretionary Access Control
ID	Identifier
IDS	Intrusion Detection System
IT	Information Technology
MAC	Mandatory Access Control
ME	Minimal Entity
RBAC	Role Based Access Control
RH	Rule Handler
TN	Trusted Network

Acknowledgements

If there is effort, there is always accomplishment.

Superior technique overcomes power. (Kanō Jigorō)

Deutsch

Diese Dissertation beinhaltet die Ergebnisse meiner eigenen Forschung. Nichtsdestotrotz wäre es mir, wenn überhaupt, nur schwer möglich gewesen, sie zu beenden, wenn nicht zahlreiche großartige Menschen mich bei meiner Arbeit unterstützt hätten.

Mein Dank gilt Professor Max Mühlhäuser, der es mir ermöglichte, meine Forschung an der TK zu beginnen und mir die Freiheit gewährte, meine Forschung selbstständig zu verwirklichen.

Besonders danke ich auch meinem Zweitprüfer, Professor Lujo Bauer, der durch sein großes Interesse und seine fachliche Kompetenz meine Arbeit sehr positiv beeinflusst hat.

Zu Beginn waren es vor allem Melanie Hartmann, Sebastian Ries und Daniel Schreiber, die mich zusätzlich unterstützten und mich durch ihren Rat und ihre positive Energie bestärkten.

Ich danke auch meinen Kollegen für die schöne und produktive Zeit des gemeinsamen Arbeitens. Besonders seien hier Leonardo Martucci, Marcus Ständer und Stefan Georg Weber erwähnt.

Leonardo Martucci und Marcus Ständer möchte ich zusätzlich herausstellen. Die zahlreichen Diskussionen haben mich bereichert und mir neue Perspektiven eröffnet. Danke für Eure Freundschaft und

moralische Unterstützung. Ich hoffe, dass wir uns nie aus den Augen verlieren.

Für die kompetente sprachliche Überprüfung zum Ende dieser Dissertation, die vielen gemeinsamen philosophischen Diskussionen und seinen ausgezeichneten Humor danke ich Daniels Umanovskis.

Über allem steht aber meine Familie, wobei ich besonders meine Eltern, Maria und Wolfgang Beckerle, herausstellen möchte. Ihr seid tolle Eltern und wart für mich stets eine Quelle der Zuversicht und der Stärke. Vielen Dank dafür und für Eure unbedingte Liebe.

English

This dissertation consists of research done by myself. Nonetheless, without the support of a large amount of great people, it would have been difficult or even impossible to finish my work.

My gratitude goes to Professor Max Mühlhäuser, who enabled me to start my research at the Telecooperation Group and gave me the freedom to realize my research independently.

I especially thank my second advisor Professor Lujo Bauer, who had a very positive influence on my research with his great interest and excellent expertise.

At the beginning of my research Melanie Hartmann, Sebastian Ries, and Daniel Schreiber encouraged me with their advice and positivity. I also thank my colleagues for the pleasant and productive time of our cooperative work.

I want to especially mention Leonardo Martucci, Marcus Ständer, and Stefan Georg Weber in this regard. Leonardo Martucci and Marcus Ständer deserve separate recognition. The numerous valuable discussions opened new perspectives and helped a lot. Thanks to you for your friendship and moral support. I hope we never lose sight of each other.

For the valuable linguistic review at the end of my dissertation, the many philosophical discussions, and for his excellent humor, I thank Daniels Umanovskis.

Above all, I want to thank my family, especially my parents Maria and Wolfgang Beckerle. You are great parents and have always been a source of confidence and strength for me. Thank you very much for that and for your unconditional love.

Remarks and List of Publications

This dissertation is based on my peer reviewed publications and on deliverables or chapters of deliverables written for the European project SmartProducts¹ that I, Matthias Beckerle, authored. Some of the deliverables are restricted or confidential. However, only content from deliverables that I exclusively authored is used in this dissertation.

I have the permission from Dr. Leonardo A. Martucci, Dr. Sebastian Ries and Prof. Dr. Max Mühlhäuser to use the parts of the peer reviewed publications they coauthored. The scientific contribution of this dissertation is exclusively done by myself.

Citations and references to these publications are not further marked in this Dissertation.

I use “we” in the following as a polite locution to include you, the reader although my own contributions are described.

List of Publications:

1. Matthias Beckerle, Diplomarbeit Interaktives Regellernen, TU-Darmstadt, March, 2009 [[Beckerle, 2009a](#)].

¹SmartProducts was an Integrated Project funded by the European Commission’s 7th Framework Programme conducted by ten partners from industry and academia. It started in 2009 and finished in 2012 [<http://www.smartproducts-project.eu/mainpage/vision>].

2. Matthias Beckerle, Section 5.15, and 5.16, Deliverable D1.2.1: Initial Concepts for Smart Products [Restricted], SmartProducts Consortium 2009-2012, July, 2009 [[Beckerle, 2009c](#)].
3. Matthias Beckerle, Section 5.5, and 6.7, Deliverable D4.1.1: Requirements Analysis for Storing, Distributing, and Maintaining Proactive Knowledge Securely [Restricted], SmartProducts Consortium 2009-2012, July, 2009 [[Beckerle, 2009d](#)].
4. Matthias Beckerle, Towards Smart Security for Smart Products, In: Smart Products: Building Blocks of Ambient Intelligence (AmI-Blocks'09), collocated with AmI'09, 18th November 2009, Salzburg [[Beckerle, 2009b](#)].
5. Matthias Beckerle, Deliverable D4.2.1: Initial Concept for Security and Privacy of Proactive Knowledge, SmartProducts Consortium 2009-2012, February, 2010 [[Beckerle, 2010a](#)].
6. Matthias Beckerle, Section 3.2, and 4.4 - 4.6, Deliverable D4.3.1: Specification of Services to Manage Proactive Knowledge [Confidential], SmartProducts Consortium 2009-2012, February, 2010 [[Beckerle, 2010b](#)].
7. Matthias Beckerle, Section 4, Deliverable D4.3.2: Initial Implementation of the Required Set of Services to Manage Proactive Knowledge [Restricted], SmartProducts Consortium 2009-2012, August, 2010 [[Beckerle, 2010c](#)].
8. Matthias Beckerle, Section 4, Deliverable D4.4.1: Evaluation Report for Initial Implementation [Confidential], SmartProducts Consortium 2009-2012, November, 2010 [[Beckerle, 2010d](#)].
9. Matthias Beckerle, Leonardo A. Martucci, Sebastian Ries. Interactive Access Rule Learning: Generating Adapted Access Rule Sets. In Proceeding of the Second International Conference on Adaptive and Self-adaptive Systems and Applications (ADAPTIVE 2010), IARIA, ISBN: 978-1-61208-109-0, pages 104-110, Lisbon, Portugal, November 21-26, 2010 [[Beckerle et al., 2010](#)].

10. Matthias Beckerle, Deliverable D4.2.2: Final Concept for Security and Privacy of Proactive Knowledge, SmartProducts Consortium 2009-2012, February, 2011 [[Beckerle, 2011a](#)].
11. Matthias Beckerle, Section 2.2, and 6, Deliverable D4.3.3: Final Implementation of the Required Set of Services to Manage Proactive Knowledge [Confidential], SmartProducts Consortium 2009-2012, November, 2011 [[Beckerle, 2011b](#)].
12. Matthias Beckerle, Leonardo A. Martucci, Sebastian Ries, Max Mühlhäuser, Interactive Rule Learning for Access Control: Concepts and Design. International Journal on Advances in Intelligent Systems, vol 4 no 3 & 4, pages 234 - 244, year 2011 [[Beckerle et al., 2011](#)].
13. Matthias Beckerle, Section 4, Deliverable D4.4.2: Evaluation Report for Final Implementation [Confidential], SmartProducts Consortium 2009-2012, February, 2012 [[Beckerle, 2012](#)].
14. Matthias Beckerle, Leonardo A. Martucci, Formal Definitions for Usable Access Control Rule Sets From Goals to Metrics. Symposium on Usable Privacy and Security (SOUPS) 2013, July 24–26, 2013, Newcastle, UK [[Beckerle and Martucci, 2013](#)]. This paper received a distinguished paper award.

1. Introduction

One of the fundamental challenges of information technology (IT) is to reflect users' usability expectations in security mechanisms. The best security mechanisms are of no avail if they are too complex for the users to use correctly. Therefore, the design of mechanisms, tools, and interfaces that combine a high level of security and usability [Beckerle, 2009b, Cranor and Garfinkel, 2005] is one of the main challenges of IT-security. If security is not usable, then this may cause security breaches that possibly lead to high costs for the industry and may cause privacy loss for casual users [Symantec, 2011].

Too many IT-security solutions tend to overstrain non-expert users. In home and enterprise environments, users want to share private or confidential information but often do not define appropriate rules for access control, such as file sharing access rights. However, the imposition of such security features often leads to insecure or impractical measures, such as access control rules that are too general. In addition, users tend to deactivate security mechanisms or render them useless by granting access to everyone. This kind of behavior is very common nowadays, and has been investigated in the context of browser cookies, virus scanners, and file access controls [Herzog and Shahmehri, 2007].

Access control mechanisms are used to ensure that access to resources is granted only to authorized parties, and to ensure that authorized parties are not denied access to resources that they should have access to. Access control for digital resources such as data files has a lot in common with access control for physical locations. Access control decisions are taken according to access control policies, which can intuitively be

viewed as collections of high-level statements [Samarati and di Vimercati, 2000] that are commonly expressed as access control rules. An access control rule can be defined as a Boolean decision (ALLOW or DENY) which is taken upon the arrival of an access request. A collection of such rules is called an access control rule set.

The authoring of access control policies and their management and implementation is not restricted to specialists in computer security. Access control policy creation and administration are expected even from novice users [Egelman et al., 2011]. Social networks, for instance, require users to manage access control rules for granting access to their private messages and photos, regardless of their expertise in security. Another application that requires users to create and manage access control rules is home data sharing [Mazurek et al., 2010]. However, the task of generating and managing access control rule sets is not trivial [Bauer et al., 2009, Egelman et al., 2011, Smetters and Good, 2009]. Errors in access control rule sets can lead to unintended results, such as sharing more (or less) data than desired, and the generation of too complex access control rule sets [Smetters and Good, 2009]. Complex access control rule sets are difficult to manage and tend to have more errors.

It is indispensable to research new possibilities to support users in generating proper access control rule sets. Therefore the term “usable access control rule sets” is introduced, denoting access control rule sets that

- (i) reflect access control policies correctly and
- (ii) are easy to manage and understand according to expert opinions

As one step to achieve usable access control rule sets, this dissertation presents a novel approach that supports generation of sound and manageable access control rule sets. That is achieved by defining a set of goals for building usable access control rule sets. Those goals are then formalized. The formalization makes it possible to compare, an-

alyze and optimize access control rule sets manually or automatically and therefore provides the needed help to experts and casual system administrators.

1.1. Motivation

Usable security and usable access control in particular is important and is going to be even more important in the future. This claim can be verified by looking at the concept of privacy and challenges being raised by future smart products.

For example, in 1967, Alan Westin defined privacy as “the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others” [Westin, 1967]. The right to privacy has been firmly established in many cultures through laws for many years, but needs to be periodically adjusted to reflect new developments. For example, in January 2012 the European Commission proposed a comprehensive reform of the EU’s 1995 data protection rules (Directive 95/46/EC) to regulate the processing of personal data and strengthen online privacy rights [European Commission, 2012].

As seen in the definition by Westin, people and societies have different expectations and understandings about privacy. Identity Management [Martucci et al., 2011], Privacy preserving identifiers [Martucci, 2009] and anonymous communication [Syverson et al., 1997, Reiter and Rubin, 1997, Martucci et al., 2006] represent some of the ways to preserve privacy by focusing on decoupling the information from the user. An additional approach is to control the information flow by only sending information to authorized entities.

Smart products are a class of upcoming devices that are currently being developed to bridge the gap between the real and the virtual world. They provide a natural and purposeful product-to-human interaction and context-aware adaptivity. An example for a smart product is the

Cocktail Companion that guides the user in choosing and mixing cocktails (see Section 3.4.1 and [Kasten et al., 2012])

To fulfill their tasks, smart products need knowledge about the application, the environment that they are immersed in, and confidential user data, such as the user's preferences and behavioral information. Moreover, smart products often need to exchange private / confidential information among themselves to complete collaborative tasks that require data from multiple sources, such as booking flight tickets or hotel rooms. Smart products can exist in highly dynamic environments. However, the amount of possible security and privacy breaches is proportional to the sheer number and variety of smart products. Equally, the variety of devices with different user interfaces also increases the complexity of administrative tasks for the end users.

The administration of security features in computational systems by non-expert users will become even more challenging in the future, because smart products for instance, add more complexity to such scenarios by increasing the administrative burden on the end-users. Therefore, one of the main challenges regarding smart products is the design of mechanisms that combine a customizable level of usable security.

In any regard, it is important to enable people to enforce their right to individual informational self-determination and to design and develop the necessary tools and mechanisms to support them.

Access Control

One class of such tools is access control mechanisms. Access control mechanisms, if correctly used, can help preserve users privacy and confidentiality. Usability is a key aspect there. Only if the relevant tools are easily usable they can effectively support users in enforcing their expectations about security and privacy.

The goal of this dissertation is to help users generate usable access control rule sets that reflect their expectations and allow them to control their confidential data and to allow users to preserve their privacy.

1.2. Research Questions

Three research questions are identified and answered in this thesis:

1. How can usable access control be integrated into future products?
2. What are the requirements for usable access control rule sets?
3. What are formally founded quantifiable measurements for those requirements and how can these measurements be used to support users in generating usable access control rule sets?

Research question (1.) is answered in Chapter 3 (and partly in Section 2.3), research question (2.) is answered in Chapter 4, and research question (3.) is answered in Chapter 5.

1.3. Scientific Contributions

The scientific contributions are directly related to the research questions presented in the previous Section 1.2.

They are:

1. An analysis, goals, concepts, and a design towards usable access control in smart products
2. Six requirements for usable access control rule sets
3. Sets and quantitative metrics of the six requirements for usable access control rule sets

Each scientific contribution can be found in the same chapter as its respective research question: (1.) in Chapter 3 (and partly in Section 2.3), (2.) in Chapter 4, and (3.) in Chapter 5.

1.4. Research Method

The research method is based on nine steps in three blocks as seen in Table 1.1.

Table 1.1.: Methodology

Block 1

1.a	Analysis of how core security objectives can be achieved
1.b	Integration of access control into smart products
1.c	Proof of concept and functional tests

Block 2

2.a	Usable access control requirements engineering
2.b	A pilot study
2.c	Refinement of requirements

Block 3

3.a	Formalization of requirements
3.b	Deduction of linear metrics from requirements
3.c	Evaluation with users and experts

In the first part we analyze solutions for usable security and show how they can be integrated into smart products. Ideas, concepts as well as a design are presented.

The second part consists of a pilot study with system administrators (using semi-structured interviews) and an analysis of papers mainly presented at the scientific conferences CHI and SOUPS. That leads to the informal definition of six goals for building usable and secure access control rule sets.

In the third part we formalize these goals and assign a set (that contains elements that influence the usability of the rule set negatively regarding the goal) and a metric to each. The introduced metrics allow the assignment of a weighted score to each goal. Assigning scores to rule sets allows users to evaluate them, identify weaknesses and compare alternatives, allowing for automation of the optimization process in future work. We evaluated the helpfulness of these metrics to users in creat-

ing usable access control rule sets, and how our metrics correlate to the opinion of IT support professionals.

1.5. Thesis Structure

In this dissertation, Chapter 2 presents the background and the related work on usable access control. We analyze how core security objectives can be achieved in the context of smart products as an example. Therefore, the usability aspects of current security tools are analyzed. This analysis shows that security mechanisms can often be made transparent to the user by automating them. Exceptions exist for access control mechanisms in the context of usable access control rule set configuration.

In Chapter 3 it is shown how access control can be integrated into future products by showing an example of this process for smart products. Ideas, concepts, and a design are presented.

In Chapter 4 informal requirements for usable access control rule sets are defined. After an analysis of the related work and a pilot study that was carried out with experts, six requirements for usable access control rule sets are refined.

By formally defining these six requirements and the related sets and metrics in Chapter 5, it is possible to analyze and compare different rule sets concerning their security and usability.

In Chapter 6 the formalization is evaluated. It is shown that the sets and the related metrics help system administrators and casual users to generate more usable access control rule sets, and that the metrics directly reflect the opinion of experts.

In Section 7 an outlook is given and the future vision of Interactive Rule Learning is presented. The dissertation is completed with final remarks and conclusions in Section 7.5.

2. Background

In this chapter we present the background of our research into usable access control. It starts with some general explanations about pervasive computing in Section 2.1 where smart products are introduced as an example domain for the following Sections. That section is followed by security objectives of IT-security in Section 2.2 and how they can be achieved in the context of the smart products domain in Section 2.3. In Section 2.4, a detailed look into access control mechanisms is given, whereas Section 2.5 explains what rule learning is. The background chapter concludes with an analysis of the state of the art in usable access control and a summary.

2.1. Pervasive Computing

Pervasive computing is a reality today. The vision of Mark Weiser [Weiser, 1991] has become true. Computers are all around us, and mostly invisible. They can be found in cars, nearly every electronic device, and start to be integrated in a lot of non-electronic objects like clothes. It opens up a lot of new possibilities. At the same time these devices gather data and are often connected among themselves and with the internet. It is indispensable to control their flow of information with proper access control mechanisms and fitting access control rule sets.

2.1.1. Smart Products

Smart products are a new class of devices that are currently being developed to bridge the gap between the real and the virtual world. They provide a natural and purposeful product-to-human interaction (usability) and context-aware adaptivity (knowledge about the environment that influences their behavior). To fulfill their tasks, smart products need knowledge about the application, the environment that they are immersed in, and confidential user data, such as the user's preferences and behavioral information.

Examples for smart products can be found in [Kasten et al. \[2012\]](#) and a smart product named the Cooking Companion is described in Section [3.4.1](#) and a video about it can be found online¹.

The official SmartProducts website [[SmartProducts](#)] describes smart products with the following words: “Smart products help customers, designers and workers to deal with the ever increasing complexity and variety of modern products. Smart products leverage ‘proactive knowledge’ to communicate and co-operate with humans, other products and the environment.”

Proactive Knowledge

Proactive knowledge is described with the following definition: “Proactive knowledge encompasses knowledge about the product itself (features, functions, dependencies, usage, etc.), its environment (physical context, other smart products) and its users (preferences, abilities, intentions, etc.). In addition, proactive knowledge comprises executable workflows and knowledge about interaction, enabling the smart product to proactively engage in multimodal dialogues with the user. Thereby, smart products ‘talk’, ‘guide’, and ‘assist’ designers, workers and con-

¹SmartProducts Demonstrator: CocktailCompanion
<http://www.smartproducts-project.eu/index.php/videos>
<http://www.youtube.com/watch?v=LNxUGPfxqU>

sumers dealing with them. Some proactive knowledge will be co-constructed with the product, while other parts are gathered during the product lifecycle using embedded sensing and communication capabilities. The outcome of SmartProducts will impact the manufacturing and consumer domain, primarily targeting consumer goods, automotive and aerospace industries, spanning both product innovation (for consumer goods and automotive) and process innovations (for automotive and aerospace)” [SmartProducts] (see Figure 2.1.1).

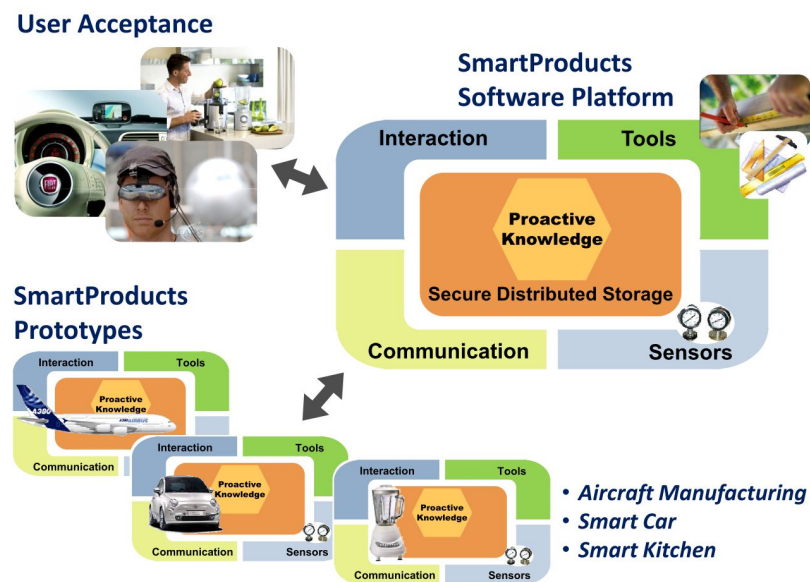


Figure 2.1.1.: SmartProducts Overview [SmartProducts]

Moreover, smart products often need to exchange private / confidential information among themselves to complete collaborative tasks that require data from multiple sources, such as providing users with guidance on cooking food. Smart products can be a part of highly dynamic environments. A general introduction to smart products can be found in [Mühlhäuser, 2008].

More information about the SmartProducts high level architecture can be found in [Schreiber et al., 2011].

2.1.2. Security for Pervasive Computing

The amount of possible security breaches is proportional to the number and variety of smart products. Correspondingly, the variety of devices with different user interfaces also increases the complexity of administrative tasks for the end users. Therefore, one of the main challenges of IT-security regarding smart products is the design of mechanisms that combine a high level of security and usability [Beckerle et al., 2010, Beckerle, 2009b, Cranor and Garfinkel, 2005].

Usability is even more important when implementing security in highly connected devices that exchange private information and are maintained by casual users, compared to traditional security systems that are managed by experts.

More details about usable security in SmartProducts as an example for pervasive computing can be found in Section 2.3 and Chapter 3.

2.2. Core Security Objectives

In this Section, we introduce some key security objectives that are going to be used throughout this dissertation: confidentiality, integrity, authenticity and authorization. The definitions are based on the definitions found in [Eckert, 2009].

2.2.1. Confidentiality

Confidentiality means that the assets of a computing system are accessible only by authorized parties. Confidentiality is usually implemented using cryptographic algorithms.

There are symmetric encryption mechanisms, such as AES [Daemen and Rijmen, 1999], and asymmetric ones like RSA [Rivest et al., 1978]. Encryption demands the distribution of cryptographic keys among participating devices. Asymmetric key encryption performs slower than

symmetric key encryption but has the advantage of having separate keys for encryption and decryption, allowing one key to be public while maintaining security. Hence, large amounts of data are rarely encrypted using asymmetric keys, instead only select data is, such as keys for symmetric encryption.

2.2.2. Integrity

Integrity means that assets can be modified only by authorized parties or only in authorized ways. Integrity is mostly implemented using one-way functions in combination with cryptographic algorithms.

Integrity has to ensure that any unauthorized change of data is recognized. Data integrity is usually accomplished using one-way hash functions and public key encryption, or just with symmetric keys. Message Authentication Codes (MAC) [[Krawczyk et al., 1997](#)] are implemented using symmetric keys and digital signatures with public-private key pairs [[Rivest et al., 1978](#)].

2.2.3. Authenticity

Authentication is required to obtain a proof of correctness over an identity claim. Authenticity means that an entity can prove who or what they claim to be. Authentication services are usually implemented by proof of knowledge, proof of ownership, or proof of biometric trait.

2.2.4. Authorization

Authorization means that policies are used and enforced to specify access rights. One way to implement authorization is through access rules, collections of which are called rule sets, that are used by access control mechanisms to determine if an entity is allowed to access information or not.

2.3. Achieving Security Objectives for Smart Products

In this section we analyze how the core security objectives (see Section 2.2) can be achieved. Such an analysis shows that there already are possible solutions that can be applied for confidentiality, integrity, and partly for authentication services to mostly automate them and thus make them transparent to the user. In such a context, we show how and why confidentiality, integrity and authenticity can be automated quite well, but authorization cannot because of individual user preferences that have to be taken into account. Smart products are used as a showcase.

The analysis regarding confidentiality is presented in Section 2.3.1, regarding integrity in Section 2.3.2, regarding authenticity in Section 2.3.3, and regarding authorization in Section 2.3.4. In each section, we analyze the usability aspects of related security solutions. This initial analysis is used to identify usability gaps in basic security mechanisms.

2.3.1. Approach for Confidentiality

Symmetric cipher algorithms are fast and widely used to assure confidentiality. For a symmetric cipher algorithm, a pre-shared secret is required. All participants of a communication need this pre-shared secret (key) to decrypt and encrypt the data. In smart products, the process of symmetric key distribution is a potential challenge because, if a unique key is required for every pair of communicating entities, the amount of keys required is $\binom{n}{2}$, where n is the total number of communicating devices.

Fortunately, the required en- and decryption keys can be exchanged with public key cryptography like RSA [Rivest et al., 1978], which is an asymmetric cipher algorithm. It is feasible to embed public-private key pairs into smart products, which would reduce total number of keys to

2n. Such an approach is sufficient in principle and implements confidentiality into highly dynamic environments using existing and standard cryptographic systems.

The encryption itself is performed automatically and should not have a significant negative impact on usability.

2.3.2. Approach for Integrity

Since cryptographic tools like RSA are expected to be embedded into smart products (as seen in Section 2.3.1), there are going to be cryptographic tools available for securing data integrity. The process of proving data integrity can be fully automated and does not need user interaction.

2.3.3. Approach for Authenticity

In smart product scenarios there are three basic types of authentication: device-to-device, device-to-user, and user-to-user. There are sufficient mechanisms based on digital certificates that can carry out device-to-device authentication automatically. Device-to-user and user-to-user authentication can be implemented using proofs of knowledge, biometric traits or digital tokens together with public-key encryption. In such a case, once users have authenticated themselves one time to a smart product, the device might be used to automate future authentication procedures between users and other devices.

2.3.4. Approach for Authorization

Smart products are delivered with predefined default rule sets. Such approaches, however, disregard adaptivity to the end-user. The general problem results from the diversity of user preferences, so more information regarding the users is required. Authorization problems regarding adaptivity and the user in smart products are discussed in Section

2.4, where the existing access control models are outlined and evaluated with regard to their suitability for smart product scenarios.

2.3.5. Conclusion and Analysis

For the protection goals confidentiality, integrity, authentication, and authorization, a variety of working mechanisms exist that are suitable for our smart products example. The correct configuration of these mechanisms is, especially with regard to casual users acting as administrators, an open challenge. The protection goals Confidentiality, Integrity, and Authentication can be and are already achieved with a minimal number of user interactions by using cryptographic methods and biometric authentication. For authorization, the task of configuring access control mechanisms involves a significant amount of often difficult user interactions for defining proper access rules. Therefore, this dissertation focuses in Chapter 4, 5, and 6 on the configuration task of access control rule sets.

The next Section will analyze different access control models regarding their suitability to smart product scenarios

2.4. Access Control Models

The role of AC mechanisms, which are implemented in accordance to AC models, is to ensure that only authorized entities are able to access the information and functions of a computer system (principle of authorization) [Stajano, 2002].

This section provides an overview of different access control models and an evaluation of such models regarding their suitability to smart product scenarios. In this section, we describe the following access control (AC) models: Blacklists, Mandatory AC (MAC), Discretionary AC (DAC), Role-Based AC (RBAC), and Attribute-Based AC (ABAC). This section concludes with a set of recommendations for an AC model

suitable for smart product scenarios. It concludes that ABAC models together with Blacklists is the most suitable solution for such scenarios.

2.4.1. Blacklists

A Blacklist is a very simple AC model that blocks all requests from entities that are included in a list. It is used to thwart known or recurrent attackers. An examples for blacklists are anti-spam filters for email accounts where mails from known spam providers are automatically blocked. Blacklists have to be configured manually, although sometimes they can be updated automatically according to predefined rules, e.g., multiple unauthorized requests, or a series of failed authentication procedures. Blacklists are usually faster than other AC mechanisms because their complexity class is lower, its performance can be $\mathcal{O}(1)$ with a very small constant factor for the blacklist lookup. Blacklists are rather simple to implement and use but also rather inflexible, as no conditional access policies can be defined.

2.4.2. MAC / DAC

MAC and DAC are two early AC models [Brand, 1985]. They can be seen as complementary approaches.

In MAC, a central administrator controls the access rights of each entity of the system. No other entity is able to change the access rights. In such a context, Multi Level Security (MLS) (such as Bell-La Padula [Bell and Padula, 1976]) is a commonly used approach. In MLS, each entity or object of the system has a security level assigned by a central authority. Each entity is only able to access other entities or objects that have the same or lower security level. Mandatory Integrity Control (MIC) is a similar approach and is used in Microsoft Windows Vista (and later). Processes can only write or delete other objects with a security level lower or equal to their own.

DAC differs from these approaches in that each owner of a file can grant access rights to other entities. That way, users are able to share objects among each other [[Sandhu and Samarati, 1994](#)]. DAC is used for example in UNIX for sharing data and resources, where the owner of a file is able to set the corresponding access rules.

2.4.3. RBAC

RBAC [[Ferraiolo and Kuhn, 1992](#)] introduced a new approach by assigning roles between the entity and its related rights. Access rights are therefore assigned to roles rather than users. Each entity can have several roles and each role can be held by multiple entities. For administrative purposes, roles are established first, and afterward they are assigned to entities. Since roles rarely change, this significantly reduces the complexity for administering RBAC after the initial setup. Roles can change dynamically, thus the user might gain and lose roles automatically when doing particular tasks.

2.4.4. ABAC

One of the newest models is ABAC [[Yuan and Tong, 2005b](#)]. ABAC uses attributes instead of roles to link rights to entities. For accessing a resource, entities have to verify that they have the according attributes needed for accessing the resource as demanded by the access policy. Examples for attributes are the name of an entity, the age or the nationality.

This procedure allows the use of dynamic conditions encoded in attributes, such as the location of an entity, to decide whether to grant access or not. Since the role as well as the security level of an entity can be seen as attributes, it is possible to integrate concepts known from other AC models like MAC, DAC or RBAC [[Jin et al., 2012](#)].

The formal definition of ABAC as used in this dissertation is given in Chapter 5.

2.4.5. Hybrid Approaches

In reality, the distinction between different AC models is not as strict as shown in this section. There are hybrid models like the Location-Aware Role-Based Access Control (LRBAC) [Ray et al., 2006], which allows the use of a geographical location as a “role”. It is often possible to derive a less complex AC model from a more complex one, e.g., it is possible to create a MAC, DAC or RBAC mechanism from an ABAC model.

Other approaches extend access control mechanisms or models with additional functionality like RBAC with trust, that integrates trust and RBAC for cloud data storage [Zhou et al., 2014] or access control models with break-glass [Brucker and Petritsch, 2009].

2.4.6. Access Control Model Conclusion

The access control model considered in this work is the attribute-based access control (ABAC) model [Yuan and Tong, 2005a]. In contrast to role-based access control (RBAC) [Ferraiolo and Kuhn, 1992] which uses roles to associate access rights to users, ABAC uses attributes. ABAC is more flexible than RBAC since dynamic conditions can be encoded in attributes. Moreover, ABAC can be used to implement other mechanisms, such as RBAC. Therefore, a formalization based on ABAC also covers many other access control mechanisms.

2.5. Rule Learning

Rule learning is a sub-field within machine learning. It generates rules that reflect a set of observations or instances. Since the optimization of AC rules is the focus of this dissertation, it seems worthwhile to look into rule learning, which is especially useful for automating the process of future access control rule set optimizations and can be used as a mechanism for intrusion detection systems (see Section 3.1.1).

2.5.1. Machine Learning

There are many definitions for Machine Learning with a considerable degree of variety:

- Definition 1: “Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time.” [Simon, 1983]
- Definition 2: “The study and computer modeling of learning processes in their multiple manifestations constitutes the subject matter of machine learning.” [Carbonell et al., 1983]
- Definition 3: “Learning is constructing or modifying representations of what is being experienced.” [Michalski, 1986]
- Definition 4: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience.” [Mitchell, 1997]

While the first definition of machine learning reduces it to improvement in the performance of a specific task or tasks, the second definition deals only with the description and modeling of machine learning. Definition 3 again describes only the learning per se, without going into machine learning. The fourth and final definition from T.M. Mitchell provides the definition that is used in this dissertation. It is measurable and contains both the aspect of learning as well as the aspect of problem solving.

The following description, which the fourth definition flows into, is used in this dissertation:

Machine learning is dealing with computer programs that can learn by experience. Just as the science of biology has different approaches to describe learning, such as behaviorism or cognitivism [Bimmel et al., 2000], there are also different approaches in machine learning, to understand learning and to implement it.

A distinction is made between unsupervised learning and supervised learning.

“Supervised” means here that there is already a set of observations and for each observation in this set the result is known. For example, a number of a person’s previous purchases was observed and now it shall be predicted whether that person will spend, for a particular purchase, more or less than € 50. Because of previous observations a theory may be set up that makes it possible to predict the current purchase. Example: It’s Saturday, and on those days the person has always spent more than € 50, therefore it is predicted that the person is going to spend more than € 50 today.

“Unsupervised learning” means that earlier observations are not assigned to a result. You may not know, for example, what the names of different fruit lying on a table are. However, due to the characteristics of the fruit, you can still sort them by their attributes like shape and color. This approach illustrates the method of unsupervised clustering [[Hinton and Sejnowski, 1999](#)].

Over the years a range of algorithms has been developed that imitate nature or have more technical approaches as a starting point. Some, such as multilayer perceptrons [[Riedmiller, 1994](#)] or Boltzmann Perceptron Networks [[Yair and Gersho, 1990](#)], try to reproduce the functionality of a brain at the level of neurons. Other approaches, such as support vector machines [[Schoelkopf et al., 1999](#)] have an abstract mathematical approach as a starting point. The algorithms differ significantly not only in the approach taken, but also in the applications, working speed, accuracy and more [[Jin, 2005](#), [Haykin, 2008](#)].

2.5.2. Rule Learner

A rule learner is a learning algorithm that is able to generate rules out of a set of examples (instances) [[Fürnkranz, 1999](#)].

Rule-Learners pursue a very intuitive approach, compared to the algorithms mentioned in the previous section. Rule Learners try to find causalities in sets of observations and express the findings in simple rules. One such rule could be, for example: “If an animal can fly and has feathers, it is a bird”. That approach has the particular advantage of being relatively easy to understand by humans. Contrast this with for instance the hyperplane of a support vector machine, which is usually much more difficult to understand, if at all possible. This property of rules is both a psychological and a practical advantage. A psychological advantage because people are more accepting of something they can understand, and a practical advantage because it is possible to detect unintended rules quickly, and define one’s own rules easily.

2.6. State of the Art for Usable Access Control

Recent studies presented at CHI and SOUPS present challenges and discuss solutions for managing access control mechanisms. They stress that usability is fundamental for setting up manageable and secure access control rule sets. In this section we summarize their findings.

Smetters and Good [[Smetters and Good, 2009](#)] study the level of control necessary for users by examining access control policies created by users in a medium-size corporation. The access control policies regulate access to data files that are stored in a document sharing system. The system supports the creation of groups of users and implements RBAC. The authors conclude that users rarely change access rights of files or folders, and tend to store files in folders that have the appropriate access control policy, as the files would inherit the folder’s access rights. Further, users have complex access control policies resulting from the creation of access control rules with unexpected effects and of redundant rules that could be made much simpler [[Smetters and Good, 2009](#)].

The particular needs and practices of access control in home environments are analyzed by Mazurek et al. [Mazurek et al., 2010]. Home environments are usually managed by users with limited or no knowledge regarding the risks and technologies behind access control mechanisms. Hence, they describe a contrasting scenario in comparison to Bauer et al. [Bauer et al., 2009], as participants have no previous theoretical and practical experience with access control mechanisms. We highlight two of their conclusions regarding home users. First, home users desire access control mechanisms with greater granularity (complexity) because they allow to express complex scenarios more directly. Second, users wish to have short and simple policy specifications. In this dissertation we discuss those two apparently conflicting goals and how they both can be achieved.

Errors in access control settings are evaluated by Egelman et al. [Egelman et al., 2011]. The authors examine how users implement access control policies with the limited settings offered by Facebook. The participants are Facebook users recruited from a higher education institution. The paper demonstrates that users are likely to introduce errors in their access control rule sets, likely resulting in less restricted access control policies than desired. The paper also emphasizes the importance of offering feedback and guidance on how to correct access control rule sets. Feedback with no guidance is proven to result in an increased number of incorrect rules [Egelman et al., 2011].

Detection and resolution of conflicting access control rules has been studied by Reeder et al. [Reeder et al., 2011]. In particular, the authors target the problems of visualizing conflicts in access control rule sets in Windows-based operating systems. They point out two particular weaknesses in the Windows conflict resolution method arising from *deny precedences*² and *two-dimensional conflicts*³. The authors propose

²DENY rules take precedence over ALLOW rules.

³Conflicts that cannot be solved using the *specificity precedence* method. This method states that rules applied to more specific entities have precedence over rules applied to less specific entities, i.e., user-related rules have precedence over group-related rules [Reeder et al., 2011].

more suitable methods to solve the aforementioned weaknesses, along with a grid-like user interface [Reeder et al., 2008]. The interface is used to show and manipulate permissions in a more intuitive way than in the Windows standard interface [Reeder et al., 2011].

Dynamic creation of access control rules for computer file access is analyzed by Mazurek et al. [Mazurek et al., 2011]. The objective of their work is to evaluate the usability and general interest in a reactive access control mechanism. In a reactive access control mechanism, users who own data files receive email requests from others wanting to access these files. Ad hoc decisions are taken by the file owners. Decisions are either to *ignore*, *allow* or *deny* the received request. File owners could also make ALLOW and DENY decisions permanent or temporary for the current request. Reactive access control can be potentially annoying, as pointed out by the authors. The (albeit limited) monetary incentive (\$0.25/answer) and, more importantly, the limited time period (one week) and relatively low and constant load of requests used in the evaluation (15 requests/day) may have concealed some results regarding the true annoyance of a reactive access control mechanism. Furthermore, the creation of ad hoc access control rules can result in unmanageable access control rule sets regarding our goals for usable access control rule sets as described in Section 4.2.

2.7. Summary

In this chapter the background knowledge required to understand this dissertation is presented. Starting with the vision of pervasive computing and the European project SmartProducts as an example, it continues by describing the core security objectives: confidentiality, integrity, authenticity, and authorization.

Access Control mechanisms are identified as mechanisms that often need significant manual configuration because user preferences have to be taken into account. A more detailed look into Access Control Mech-

anisms is given since the main focus of this dissertation is about access control rule set generation. The section on Rule Learning gives some basic terms about the related field of machine learning. The chapter is concluded with a state of the art analysis regarding usable security.

In the next chapter it is shown how usable security can be integrated into smart products.

3. Access Control for Smart Products

Smart products are used as an example for a real life scenario which has a high demand for usable access control since they are used and configured by non-experts. Therefore, smart products are used as a motivation for Chapter 4, 5, and 6 as they demand a high level of usability and usable access control rule sets are an important step to make the configuration of access control mechanisms usable.

Smart products are built in a way that help customers, workers, and designers to effectively use them. Therefore they have to maintain knowledge about their environment, their capabilities, about the users, and about interactions. This sensitive information needs to be protected. Building appropriate security mechanisms for smart products has some challenges that have to be considered when implementing usable security for smart products, such as resource limitation, cooperative attacks and casual users that require a high level of usability to succeed at administrative tasks.

Section 3.1 lists those challenges as goals and presents and evaluates a set of proposals to address them. In Section 3.2 those proposals are refined to into usable security concepts for smart products. In Section 3.3 a design is given that integrates the concepts into a framework for smart products. The chapter concludes with a proof-of-concept prototype and functional tests for the components in Section 3.4.

3.1. Challenges and Goals

In this section challenges of smart products are shown and taken into consideration to develop an adapted vision of usable security for smart products. Here we mainly focus on access control aspects. Other security issues like authentication of users, integrity, and confidentiality are only described in a very basic way (see Section 2.2 and Section 2.3). For more information about challenges regarding additional security topics related to smart products, please refer to the EU research projects MOSQUITO (<http://www.mosquito-fp7.eu/>), AWARE-NESS (<http://www.aware-project.eu/>), PRIME (<https://www.prime-project.eu/>) and PrimeLife (<http://primelife.ercim.eu/>).

In the process of the smart products requirement engineering, four aspects were identified that differentiate smart products from classical products. Proactivity (see Proactive Knowledge in Section 2.1.1) requires a high degree of automation and cooperation between smart products. Compared to home computers, they have less resources such as computational power and storage. Since they are used by non-experts, they need a high level of usability for user acceptance. These aspects can be classified and interpreted for smart products security in the following way.

For the security aspects of smart products, the following two points are considered in particular and detailed in Section 3.1.1:

- *Cooperation*: The privacy of the users has to be preserved and the confidential data stored in smart products has to stay secret. If smart products combine their capabilities, they increase the chance to detect and prevent attacks.
- *Preserve Resources*: Security mechanisms need to minimize the amount of computational resources like memory and computational power.

For the usability aspects of smart products, we focus in particular on the following two points that are detailed in Section 3.1.2:

- *Automation*: Automation should be used wherever it is possible, to minimize the amount of interaction users need to perform.
- *Reflect User Preferences*: The preferences of a user have to be reflected in the security mechanisms.

3.1.1. Security Aspects of Smart Products

The realization of security for the storage and distribution of proactive knowledge (see Section 2.1.1) leads to a high number of interactions between the users, or devices representing them, and their smart products. In addition, smart products deal with a large amount of confidential data such as personal preferences and current whereabouts. Furthermore, smart products also store confidential manufacturer information, e.g., maintenance and repair information. It should be ensured that no unauthorized entity can obtain data that is private to the user or manufacturer [Iachello and Hong, 2007].

To protect the data, one has to precisely define which entities are allowed (authorized) to do what on which data. This mechanism is called authorization (see Section 2.2.4). Therefore, it is necessary to identify the entities that want to access the data and to specify their rights. The same applies to the secure distribution of data: on which devices can the data be stored without any concerns, and which devices require special mechanisms in order to protect the data? Solving that requires an appropriate identity and rights management system like Attribute Based Access Control (ABAC, see Section 2.4.4) [Artz and Gil, 2007, Grandison and Sloman, 2000].

Preserve Resources

Limited resources of some smart products have to be considered when designing security mechanisms for them. There are three different facets to it. First, the security mechanisms need to have low resource consumption. Second, denial-of-service attacks need to be considered

especially. Third, it can be difficult for a single smart product to employ complex attack detection algorithms.

Cooperation

IT security in ubiquitous computing, and for smart products in particular, is a difficult task because more systems also imply more and potential new weaknesses. Distributed and redundant data storage allow new attacks that a single smart product cannot detect. Location information is one such example. A single query is necessary for path finding or getting local weather data. But if it happens over a period of time, it can be an attack against location privacy. Especially cooperative attacks by multiple attackers against smart products are difficult to handle, since single request by different attackers are difficult to detect as an attack. Thus, classical individual security concepts may not be sufficient for smart products.

A promising approach is to defend collaboratively, too. Therefore we recommend that smart products form a network of trusted devices consisting of all the smart products of the same owner. This allows smart products to combine their capabilities to defend against possible attacks (see Trusted Networks in Section 3.2.2).

In addition, since smart products will often operate in insecure environments, information rights management alone is insufficient for preserving the user's and manufacturer's privacy. If an attacker is able to steal the identity of a device that is already part of a trusted smart products network, or if the attacker can manipulate a device to get control over it, he can manipulate smart products with the full rights of the compromised device without being noticed. Rights management alone can only work properly if the attacker is an identifiable, discrete entity in the smart product network.

To defend against masquerading and exploitation of software and/or hardware errors, intrusion detection is a possible solution. Intrusion

detection allows finding misbehavior by comparing the actual behavior of entities to their expected behavior. Therefore, intrusion detection has the ability to detect disguised attackers as soon as they perform an attack, that includes unexpected behavior. It acts as a second line of defense after rights management which is especially important in a smart product environment because of changing participants and the variety of smart products.

3.1.2. Usability Aspects of Smart Products

The realization of security for the storage and distribution of proactive knowledge as it relates to smart products, faces a lot of usability challenges: smart products are intended to provide everyday support to users, leading to a high number of interactions, that cannot be easily handled manually. Without a high degree of automation, users are forced to configure security mechanisms of multiple devices or copy relevant data from one device to another. In addition, users have to adapt smart products to their own preferences. It is important to support the users and not to overstrain them as motivated in Chapter 1.

Automation

The most usable security mechanism is one that would automatically fulfill the user's security expectations. Focusing on usability for smart products, to get a high degree of user acceptance some particular conditions have to be taken into consideration when designing and implementing smart products. Smart products are tied into the user's everyday environment. A high degree of usability is indispensable for devices that get used on a daily basis. If security mechanisms complicate everyday tasks for users, they may try to deactivate the security as discussed in chapter 1. Therefore, one goal is to integrate security directly into the normal interactions that the user will perform anyway. For instance, it should be sufficient to take the new device home and press two keys to

add it to the network or to authenticate by touching your mobile phone as it automatically scans your fingerprints.

In terms of access control, the challenge is that, in a fully connected world, there are too many access requests to handle manually. This problem can be minimized by automating access control decisions and by automating the process of security configuration. Automating access control decisions is commonly done by today's access control mechanisms. The configuration process, however, is still a challenge and is discussed in Chapter 4, 5, and 6.

Reflect User Preferences

Some manual interaction is still necessary, especially when users have to enforce individual preferences (see Section 2.3). The whereabouts of, for instance, a taxi, might be publicly available, while the whereabouts of a lawyer or a doctor that is visiting clients should be kept private. Today, it is quite difficult to automate this. But it should be as easy as possible for the users to make devices reflect their preferences. Therefore, smart products require support mechanisms such as the usable security tools presented in Chapter 5 to reduce and simplify the necessary user interactions by giving users the possibility to define access control rules and, if possible, to generate them automatically (see Chapter 7.4).

3.2. Concepts for SmartProducts Security

In this section the previously mentioned four goals from Section 3.1 - Automation, Reflect User Preferences, Cooperation, and Preserve Resources - are integrated into the concepts of *Usable Access Control* (see 3.2.1) and *Layered and Cooperative Security* (see 3.2.2). The relationship between the basic ideas described in Section 3.1 and the concepts can be seen in Table 3.1. Figure 3.2.1 describes in addition the relationship to the design presented in Section 3.3.

Table 3.1.: Basic Goals related to Concepts

	Preserve Resources	Cooperation	Automation	Reflect User Preferences
Layered and Cooperative Security	x	x	x	
Usable Access Control		x	x	x

3.2.1. Layered and Cooperative Security

A multilayer approach is envisioned for achieving a high level of security and privacy in smart products. Since smart products are adaptive devices, an AC mechanism with maximum flexibility is desirable. For that reason, ABAC is an appropriate choice (see Section 2.4). Furthermore, for performance reasons a blacklist is used as a filter to keep misbehaving entities away from deeper layers such as the ABAC. A blacklist is simple and much faster than ABAC and especially than Intrusion Detection Systems.

A problem occurs when an attacker is able to perform an impersonation attack, i.e., steal the identity of a smart product or of an authorized user. In this way the attacker would obtain the rights of the original entity and can use these rights for further attacks. The attacker could also tamper with and compromise smart products that are part of the network, and monitor communications or propagate harmful messages in it. The goal is to detect such hostile-acting entities for instance with an intrusion detection system and to restrict their access rights or remove them entirely

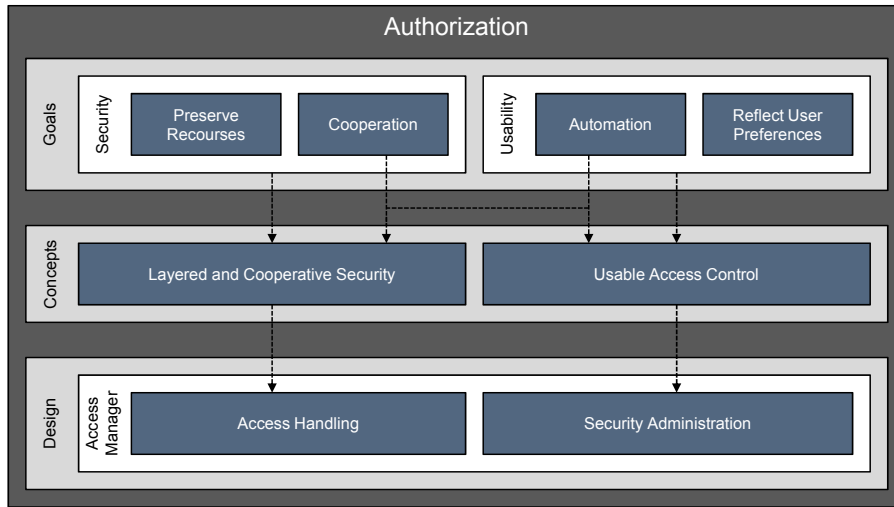


Figure 3.2.1.: Overview of Relationships between Goals, Concepts, and Design

from the network by denying further communication with them [Debar et al., 1999].

In addition to this vertical structure consisting of a blacklist, an ABAC mechanism and an intrusion detection system, we also envision a horizontal structure of defense to make the layers even stronger against potential attacks. As depicted in Figure 3.2.2, the intrusion detection systems of a set of trusted smart products detects misbehavior cooperatively. To determine the trustworthiness [Neisse et al., 2007] of smart products, we propose that trusted nodes build up a virtual network, a so called Trusted Network (TN).

In general, two classes of TNs can be distinguished: first, TNs composed of devices that have the same owner; and second, TNs that are composed of devices of the same manufacturer. Thus, every smart product participates in two TNs. TNs are strictly separated from each other, which means no confidential information is exchanged between TNs (i.e., no transitivity). For example, the manufacturer is not able to access the owner's data and vice versa. Within a single TN, smart products can exchange confidential information like access rules (owner TN

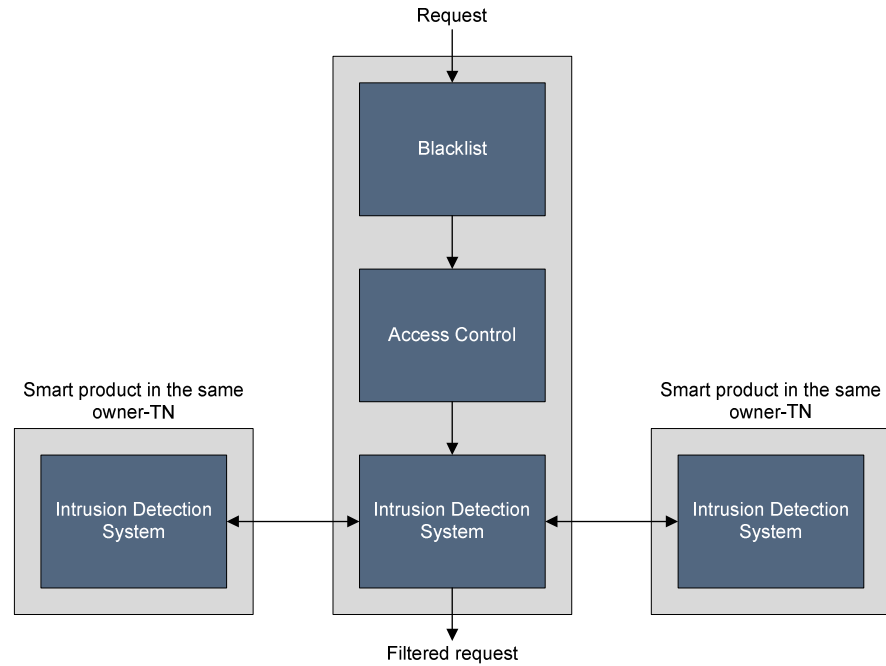


Figure 3.2.2.: Vertical and Horizontal Security Layers

or manufacturer TN), user profiles (owner TN), or manufacturer data (manufacturer TN). To implement the idea of TNs, smart products of the same TN have a pre-shared secret. This pre-shared secret is sent to every smart product after it is bought and activated for the first time. In this process, the user has to verify that this new smart product is allowed to integrate itself into the owner's TN. This can be done manually, e.g., with out of bound communication, or automatically with the help of a minimal entity (ME). Overall, this is similar to the Resurrecting Duckling Model proposed by [Stajano, 2002].

If an entity is observed to act in an improper manner (e.g., multiple attempts of accessing a forbidden resource or a denial-of-service attack) by the intrusion detection system, it will get a temporary or permanent entry in the blacklist and will be blocked as long as the entry exists. To maintain the usability of the overall security concept, an appropriate management interface is required that allows the user to edit a smart

product's blacklist. The specification of this interface is a subject for future research.

3.2.2. Usable Access Control

Access rights have to be managed in a usable manner. Since access rights in a smart product network may change over time, it is necessary for the assignment of rights to be clearly displayed and easily changeable by authorized entities [[Cranor and Garfinkel, 2005](#)].

As described in Section 1.1, usable access control rule sets are rule sets that

- (i) reflect access control policies correctly, and
- (ii) are easy to manage and understand according to expert opinions.

The concept of Usable Access Control is explained and detailed in Chapter 4, 5, and 6. These chapters focus on the administration of access control rule sets and introduce a formal approach to dealing with usability challenges. The vision is to enable Interactive Rule Learning for the generation and configuration of access control rule sets in the future (see Section 7.4).

3.3. SmartProducts Security Design

A usable access control mechanism for smart products is being implemented as a component of the SmartProducts software platform [[Schreiber et al., 2011](#)], part of the SmartProducts project funded by the European Commission's 7th Framework Programme. In this section we give an overview of the design of the Access Manager component in the SmartProducts software platform. The objective of this platform is to provide an open framework for developers to design hardware and implement applications for smart products. The Access Manager component is mainly responsible for the authentication, access control and

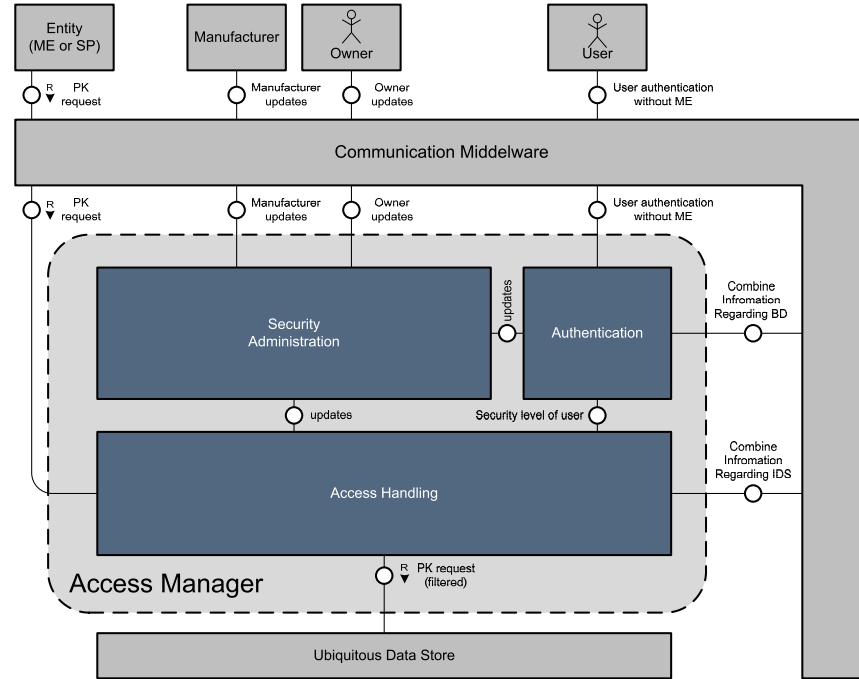


Figure 3.3.1.: Access Manager Architecture.

security administration for smart products. Figure 3.3.1 depicts the architecture of the Access Manager component. This diagram and the following diagrams in this chapter are presented using the FMC notation [Keller and Wendt, 2003].

The Access Manager has interfaces to the Communication Middleware, which handles all communications outside the device, and to the Ubiquitous Data Store, which implements an interface to one or more data stores. The Access Manager has three sub-components: Authentication, Access Handling and Security Administration. The functionality of the aforementioned sub-components is summarized below:

- The *Authentication* sub-component handles multiple authentication mechanisms, thus enabling device and user authentication such as biometric authentication. It is responsible for authenticating entities, such as users and devices. The authentication component is out of the scope of this dissertation and it is not going to be further detailed; it is only mentioned for the sake of com-

pleteness. Regardless, it is a fundamental building block of the security architecture for smart products.

- The *Access Handling* (AH) sub-component manages the black-list and the ABAC (see Section 2.4). It implements the Layered and Cooperative Security concept (see Section 3.2.1). The AH is described in detail in section 3.3.1.
- The *Security Administration* (SA) sub-component implements the rights management part of the Usable Access Control concept. It is the component where the Interactive Rule Learning (see Chapter 7.4) will take place. The SA is described in detail in section 3.3.2.

3.3.1. SmartProducts Access Handling

The Access Handling ensures that only authorized entities are able to access the proactive knowledge, which is basically a secure distributed database for Resource Description Framework data and key-value pairs¹. It filters every request through a set of rules and forwards only those requests that are deemed to be legitimate. The Access Handling component is depicted in Figure 3.3.2. This section is divided into two parts: the first part describes the access handling components, and a use case is illustrated in the second part.

Access Handling Subcomponents

The Access Handling component is composed of the Access Handler subcomponent (AH), the Blacklist Handler subcomponent (BH), the Access Control subcomponent (AC), and the Intrusion Detection System subcomponent (IDS), which are detailed next.

¹More information is available on the SmartProducts project website: www.smartproducts-project.eu.

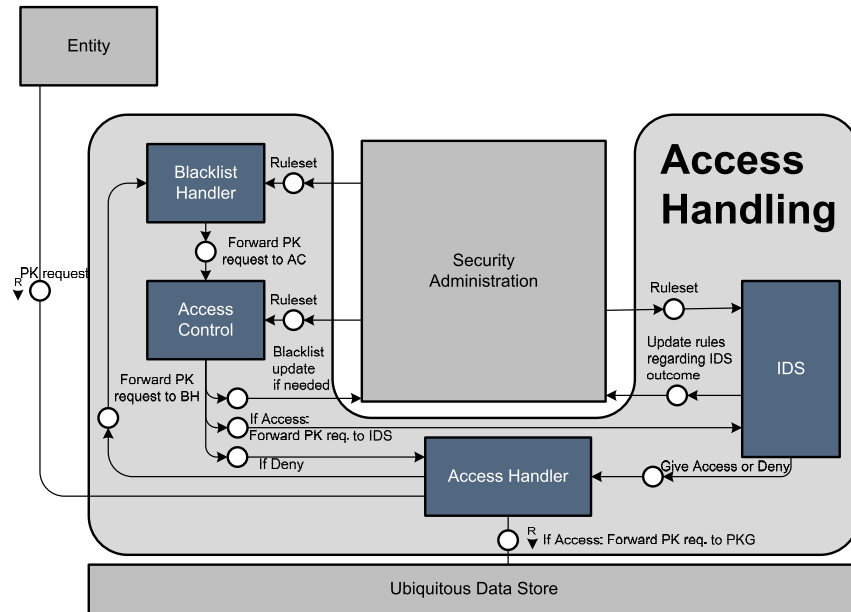


Figure 3.3.2.: Access Handling Component Overview.

Access Handler Subcomponent The AH is the interface for handling access requests. It forwards requests to the BH and informs the requesting entity about the outcome of the request. If an access request was authorized by the Access Handling component, it will be forwarded to the Ubiquitous Data Store. If it is necessary to send data back to the requesting entity to fulfill the request, the data goes through the AH.

Blacklist Handler Subcomponent The BH is the first of the three rule-based access control mechanisms. It blocks every request from entities which are listed in the General Blacklist database. If the requesting entity is not blocked, the request is forwarded to AC for further examination.

Access Control Subcomponent The AC checks if a request corresponds with the access rights of the requesting entity. All access rights are read from the AC Rules database. If an access is refused, the ID of the requesting entity will likely be added to the general blacklist

database for a period of time. If an entity often tries to access a resource it has no authorization for, it will be added to the general blacklist database. In case a request was blocked because of a blacklist entry, no information will be sent back to the requester. Otherwise the AH informs the requesting entity that it was blocked. If a rule exists that allows the requesting entity to access the requested resource the request is forwarded to the IDS.

Intrusion Detection System Subcomponent The IDS verifies that an access request does not deviate from the expected behavior of the requesting entity. The expected behavior is stored in the behavior whitelist database. If an access request is determined to be abnormal, the IDS asks the IDS of other smart products around for help. The IDSs around combine their knowledge about the requesting entity to determine if the entity is the one it claims to be. The outcome will be used to update the database for better future results and is also forwarded to the AH. The IDS should ideally be able to exchange information with IDSs on other smart products in order to cooperatively detect intrusions. The design and implementation of the IDS component is out of the scope of this dissertation.

Use Case

In this use case, a user requests information from a smart product. To decide whether a data access may take place, the respective request must pass through all subcomponents of the Access Handling component. Only when all subcomponents approve the right to access the data, the request will be forwarded from the AH to the Ubiquitous Data Store. This use case is depicted in Figure 3.3.3.

3.3.2. Security Administration

The Security Administration component contains the Rule Handler sub-component (RH) – a bidirectional interface for rule updates between the

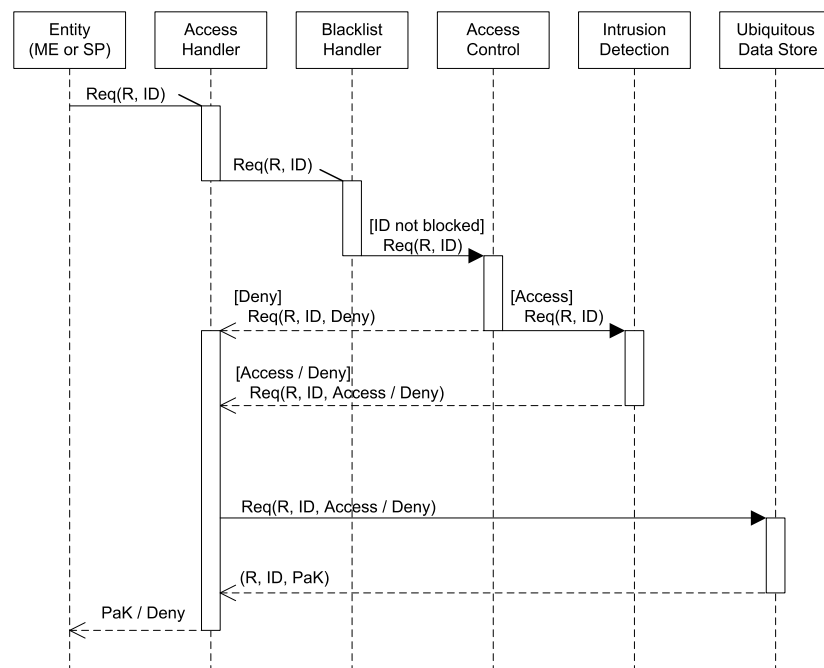


Figure 3.3.3.: Access Handling Use Case.

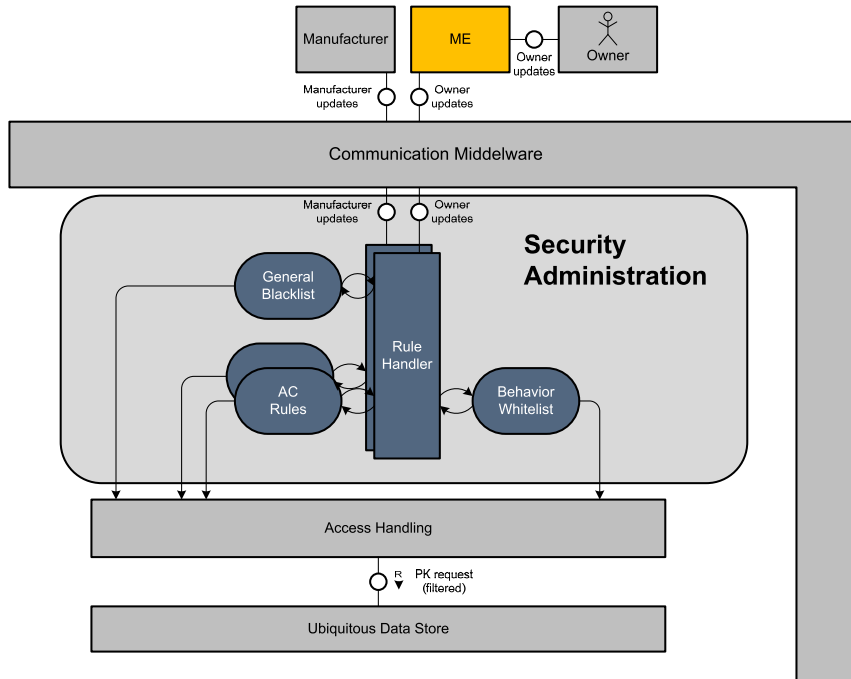


Figure 3.3.4.: Security Administration Module.

owner, the manufacturer and the Access Handling of the smart product. The RH maintains three databases for the different Access Handling subcomponent. Every smart product in the same Trusted Network [Beckerle, 2011a] of the owner has the same owner-specific access rules for redundancy and as a measure against simple manipulation. To help the user define suitable rules, a new research topic called Interactive Rule Learning will be investigated in the future (see Section 7.4). The Security Administration Module is depicted in Figure 3.3.4. This section is divided into two parts: the first part describes the Security Administration components, and a use case is presented in the second part.

Security Administration Components

The Security Administration component consists of the Rule Handler subcomponent (RH), the set of access control rules, the blacklist and

the whitelist. The RH exchanges information with the Minimum Entity (ME). A description of the RH and the ME are provided below.

Rule Handler The RH manages the three databases: the general blacklist, AC rules, and behavior whitelist. The general blacklist is used by the BH and contains the identities of blocked entities. There are two databases for AC rules: the first database contains user-defined access rules, and the second database consists of manufacturer-defined access rules that are required for maintenance (e.g., firmware updates). The behavior whitelist has rules that describe the normal behavior of entities interacting with the smart product, and is needed for the IDS.

The RH communicates with the Minimal Entity (ME) or an equivalent device to support the user's management of the different databases. This is envisioned to be done with the support of Interactive Rule-Learning (see Section 7.4 and [Beckerle, 2011a]). The manufacturer of the smart product is only able to update the manufacturer-defined AC rule database.

ME The ME is a device that represents the user in the digital world. It is used to easily authenticate the user (out of scope) and serves as a user interface for configuring the RH. Alternatively, this functionality can be integrated in a smart product. MEs are not the focus of this dissertation on are only mentioned for completeness. More information about MEs can be found in [Beckerle, 2011a]

3.4. SmartProducts Desing Evaluation

The evaluation of the design of the Access Manager takes place in two ways: a proof-of-concept prototype that is described in Section 3.4.1 and functional tests of the implementation of the SmartProducts security design in Section 3.4.2.

3.4.1. Proof-of-Concept Prototype



Figure 3.4.1.: Cocktail Companion Overview

The first part evaluates the interoperability of the access manager in a prototype named Cocktail Companion, developed by TU-Darmstadt, that functions as a proof-of-concept (see Figure 3.4.1).

The Cocktail Companion guides the user in choosing and mixing cocktails. The list of cocktails shown depends on the user's age (alcoholic or non alcoholic cocktails), the available ingredients, and the user's preferences that are stored in the user profile. Alice, Bob, and Charly as adults get all available cocktails on the screen. Denise as a child sees only the cocktails without alcohol (see Figure 3.4.2).

The user authenticates at the Cocktail Companion by using RFID tags. After a successful authentication, the Cocktail Companion sends a request for the available cocktails to the Access Manager. The Access Manager adds a list of credentials with the access rights of the authenticated user to the request and forwards it to the Ubiquitous Storage. The Ubiquitous Storage forwards it to the Proactive Knowledge Base, which collects the data with consideration to the credentials. Afterward

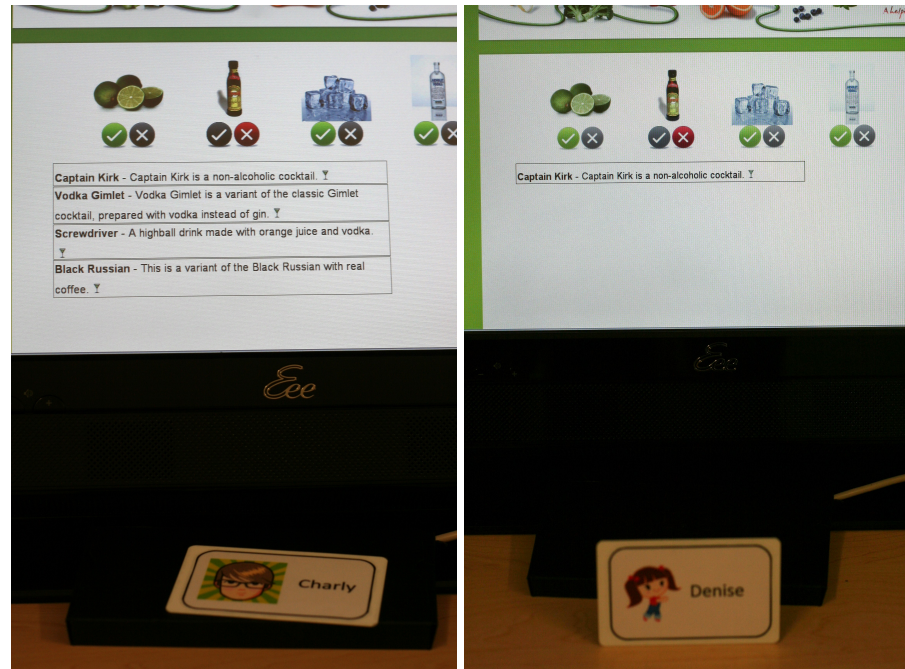


Figure 3.4.2.: Access Manager: Proof-of-Concept

the data is sent back to the Ubiquitous Storage, which forwards it to the Access Manager, which forwards it to the Cocktail Companion.

Alice and Charly have the attribute “adult”, while Denise has the attribute “child”. The applicable access control rule set describes that adults have access to all cocktails that are located in the columns “Adult” and “Child”. Denise, as a child, only has access to cocktails stored in the column called “Child” (see Figure 3.4.2).

This scenario is used for the functional tests in Section 3.4.2 (for more information regarding the Cocktail Companion and other trials see [Kasten et al., 2012]).

Table 3.2.: Access Control Functional Test Cases

ID	Test Case	Description	Wanted Outcome	Result
Sec. JUnit. 28	Get credentials for adults	Access credentials Test for Alice, Bob, and Charly. The access control mechanism extracts all relevant credentials out of the access rules and returns them.	Credentials Child and Adult are returned	Success
Sec. JUnit. 29	Get credentials for children	Access credentials Test for Denise. The access control mechanism extracts all relevant credentials out of the access rules and returns them.	Only credential Child is returned	Success
Sec. JUnit. 30	Blocking blacklisted entity	The Access Manager shall block blacklisted entities without checking for credentials. In this test case Charly is blacklisted for the purpose of testing.	Charly is blocked and gets no credentials.	Success
Sec. JUnit. 31	Get actual user	The Access Manager shall return the actual authenticated user.	The actual user is returned	Success

3.4.2. Functional Test Cases & Results

For the functional evaluation, JUnit² tests were written. The access control tests cover the main functionality of the access control: providing credentials and blocking hostile entities. The tests are using the scenario described in Section 3.4.1. Table 3.2 provides a more detailed explanation of the test cases. A brief description is given as well as the wanted outcome and the result based on the implementation. The ID refers to the ID used in the SmartProduct testing framework.

3.5. Summary

This chapter describes the basic ideas surrounding particular challenges for smart product security: automation is needed to not overstrain the user by minimizing interactions, in addition, it must be easy for the user to introduce their own preferences into the security system. Limited resources have to be considered when designing security mechanisms, especially to defend against denial-of-service attacks. By giving smart products the possibility to exchange information, a cooperative defense is still possible where single smart products would fail.

The basic ideas are integrated into two concept descriptions. The first concept focuses on security aspects and the second one on usability aspects for smart products. By using a multilayer defense system in smart products, it is possible to combine very fast mechanisms like a black-list with more sophisticated mechanisms such as ABAC and intrusion defense. This helps to defend with minimal resource usage while retaining maximum data security. To obtain a usable security system, usable access control is integrated into smart products, allowing for their easy configuration.

²see <http://junit.org/> for more details

We show how this concept can be integrated into the smart products framework by describing the design of the Access Manager that was implemented in the SmartProducts EU project.

The Cocktail Companion is described as a proof-of-concept to give an example for a usable smart product that is functionally reliant on the Access Manager. At the end of this chapter we provide functional tests of the Access Manager's functionality.

More details about SmartProducts can be found at www.smartproducts-project.eu.

The next chapter informally identifies requirements to make access control rule sets usable.

4. Informal Requirements for Usable AC Rule Sets

In the access control system for Smart Products, as well as for many types of access control systems (e.g., for Windows and Linux file system, social network access rights, home automation) access control rules are used to specify the policy that the system will enforce. This chapter will show that the usability of access control rule sets is key to the usability of access control systems that use them. Unusable access control rule sets lead to a higher level of security and usability.

In this chapter, combined with Chapter 5, we research what exact characteristics usable access control rule sets have and how they can be described. We summarize the limitations, problems and findings identified in the Background section and in our pilot study seen in the next section, and organize them into a concise set of six goals for building usable access control rule sets. This set of goals is then formalized in Chapter 5 using formal logic. Formal logic and user studies are used to corroborate the identified goals for usable access control. A formal definition can also be used to build up a computational system that can analyze, measure and offer guidance for setting up manageable access control rule sets.

4.1. Pilot Study

We started with a pilot study which consisted of semi-structured interviews with IT support professionals, i.e., experts. The objectives were

to list the usability challenges related to management of access control rule sets and to look at how the participants handled those challenges.

4.1.1. Methodology

The participants were all IT support professionals (system administrators). They were recruited from business and public sectors (universities). Seven IT support professionals from four different organizations were interviewed. All of them managed Linux- or Windows-based access control mechanisms, using tools and services like Active Directory, iptables, and firewalls. No financial incentive was offered to the participants¹.

We used semi-structured interviews as our method of inquiry in the pilot study. This method provided us the flexibility to ask for details regarding the challenges faced when managing access control rule sets. The interviews were individual and carried out under the condition that anonymity would be preserved (access control rule set details are usually confidential). All interviews were digitally recorded. We started by asking the participants about their position in the organization hierarchy and about their main tasks related to access control management.

We asked about potential problems that occur when new access control rules are defined and when existing rule sets have to be changed. Furthermore, we asked what types of errors can occur in these processes and how they are avoided or circumvented.

4.1.2. Results

All participants of our pilot study reported strict procedures for managing user rights. Changes or adjustments in the access control rule set were discussed in meetings with other system administrators. A system

¹We however promised the participants to inform them first-hand about our findings and conclusions.

administrator from an organization with about 1 000 employees estimated that one full work day is spent on such meetings every month. The administrator reported that these regular meetings were considered to be of high importance for the organization and the main objective was to guarantee the understandability and manageability of the access control rule set.

The participants also stressed the existence of two general kinds of challenges regarding the management of access control rule sets.

First, rule sets need to be restrictive and at the same time allow legitimate access:

- (G1) Rule sets have to deny unauthorized access.
- (G2) Rule sets have to grant authorized access.

i.e., all allowed accesses should be authorized and no gaps for unauthorized access should exist.

Second, rule sets needed to be understandable and manageable to help system administrators verify that the existing policies are implemented correctly². The participants reported a series of potential problem sources in access control rule sets that resulted in poor manageability. We organized those sources into the following goals:

- (G3) Redundant rules need to be removed.
- (G4) Contradictory rules need to be removed.
- (G5) Concise rule sets are better than large rule sets.
- (G6) Rule sets are easier to manage when they have been designed to facilitate the administrators' work of adding/removing users to/from rule sets.

A more detailed description of G1 to G6 that also takes related work into account can be found in Section 4.3.

The participants were also asked about the usability of different access control mechanisms. They all pointed out that indirect access control

²The distinction between policy makers and implementers identified by Bauer et al. [Bauer et al., 2009] maps directly to these two challenges.

mechanisms (like RBAC and ABAC) are more usable than direct access ones (like access control lists [[Brand, 1985](#)]). However, they acknowledged that the task of translating entity-file access decisions (e.g., user x is allowed to access file y) is more difficult in RBAC and ABAC than in the other access control mechanisms.

In the next two sections, we summarize the problems and findings identified in our pilot study and in the background Chapter 2 and describe the aforementioned six goals for building usable access control rule sets in more detail.

4.2. Definition of Goals

We define the goals in terms of ownership, objects and access control rules. Owners can grant or deny access to objects using access control rules. Objects are resources such as data files, data folders, or physical rooms. Access control rules are written in terms of ALLOW or DENY decisions. The six goals identified are:

4.2.1. (G1) Allow no more than the owner wants to be allowed.

This goal defines that a resource should be accessed only by people that are intended to have access to it. Allowing more than intended is the result of insufficient restrictive or missing access rules. Insufficient restrictive access rules are a likely consequence of errors introduced by owners as shown by Egelman et al. in their study with Facebook users [[Egelman et al., 2011](#)]. This problem is also identified in [[Smetters and Good, 2009](#)] in its analysis of documents with public access.

4.2.2. (G2) Allow everything the owner wants to be allowed.

This goal states that a resource must be available to the people that are intended to have access to it. This goal basically complements *G1*. Allowing less than the intended access is the result of too restrictive access rules. Too restrictive access rules occur when the initial access control policy is insufficient as shown in [Mazurek et al., 2010].

4.2.3. (G3) Make sure that a rule is not fully covered by another rule of the same rule set.

Redundant rules increase the complexity of an access control rule set by introducing new rules that are already covered by existing rules, thereby reducing the manageability of the access control system. Redundancies account for one of the reasons leading to errors in access control decisions [Smetters and Good, 2009].

4.2.4. (G4) Two rules within the same rule set must not conflict.

Conflicting access control rules impair the understandability of a rule set and often increase its complexity. Moreover, the resulting action from such rules will depend on the implementation of the access control mechanism's conflict-resolution method. Deny precedence implies that Deny rules take precedence over Allow rules. Allow precedence implies the opposite. The order of appearance in the rule set can be used to define the precedence too, i.e., the first fitting rule is picked. Conflict-resolution in Windows-based systems was studied by Reeder et al. [Reeder et al., 2011], who propose a new conflict-resolution method. Reeder et al. conclude that methods have inherent trade-offs as no

method is able to always deliver the desired set of permissions. In our pilot study, we confirmed the findings of Reeder et al. The IT support professionals interviewed in our pilot study stated that conflicting rules were the most annoying issue in terms of maintainability. Optimizing G4 can conflict with optimizing G5 (see Section 6.4).

4.2.5. (G5) Minimize the number of rule set elements.

Minimizing the size of rule sets reduces their complexity and facilitates visual inspection. Complexity was identified as a major problem in the manageability of access control rule sets in the user studies of Smetters and Good [Smetters and Good, 2009] and of Mazurek et al. [Mazurek et al., 2010] who evaluated distinct test environments (a medium-size corporation and home settings, respectively). After removing redundancies (G3) and (in some cases) eliminating conflicts (G4), the size of a rule set can be further optimized. One way to further optimize according to G5 is to grant rights based on attributes instead of unique identifiers (granting access rights for Students is one access rule – granting access right for individual students by using the matriculation number requires a rule for every student), by reducing the amount of attributes per rule and avoiding unnecessary rules. But unlike G3, this procedure can lead to other conflicts, e.g., opening gaps for intruders.

4.2.6. (G6) Minimize maintenance effort in a changing system.

Minimizing maintenance effort of an access control rule set with constantly changing access control policies requires the set to be manageable and understandable i.e., Goals G3, G4, and G5 indirectly help G6. Most of the changes in the rule set happen when access control policies

are modified, or when users are added to or removed from the system. Overfitting rule sets results in increased maintenance effort³.

4.3. On Goals and Derived Metrics

G1 and G2 are security related goals, as they express access control decisions. The manageability of rule sets is reflected in goals G3 to G6. All six goals for building usable access control rule sets need to be taken into account when creating new or evaluating existing rule sets. The need to evaluate all goals is a result of the non-orthogonality between the goals. Optimizing one goal might lead to a degradation of other goals in some cases, or might have a positive correlation in other cases. An example of trade-offs between goals was presented in Section 4.2 on the relation between G5 and G3.

This relationship between goals can be illustrated as follows. G2 can be maximized by defining a general Allow decision for every request. This solution conflicts with G1, as it may allow more than the owner wants to be allowed.

Reactive access control is another example that showed the relationship between our stated goals [Mazurek et al., 2011]. It allows changes to be made in the access control list according to the most current access control policy. Access control policies are defined by the owner on an ad hoc basis. Thus, G2 is influenced positively as everything the owner wants to be allowed is allowed.⁴ However, negative effects on goals G3, G4, G5 and G6 would manifest due to drawbacks in reactive access control, such as the lack of consistency checks in the resulting access control rule set, the probable creation of redundant and conflicting rules, and the potential annoyance of having to make ad hoc decisions regarding access control requests.

³We use the term *overfitting* according to its machine learning definition. In the scope of this dissertation, it means that rule sets that perform well at the current state of the system may perform poorly if the system is modified.

⁴There are no guarantees that reactive access control maximizes G2 since DENY decisions may have permanent effects.

The fulfillment of the goals can also be used to reduce mismatches between people's mental models of access control mechanisms and their actual implementation, which is a problem identified by Mazurek et al. [Mazurek et al., 2010]. Such mismatches can be reduced if users are able to verify the implemented policies and compare the actual implementation with the desired policies.

4.4. Summary

In this chapter we answered the research question: “What are the requirements for usable access control rule set configuration?”

A pilot study consisting of semi-structured interviews with IT experts was performed. The gathered information was refined into six informal goals for usable access control rule sets:

- (G1) *Allow not more than the owner wants to be allowed.*
- (G2) *Allow everything the owner wants to be allowed.*
- (G3) *A rule must not be fully covered by another rule of the same rule set.*
- (G4) *Two rules within the same rule set must not conflict.*
- (G5) *Minimize the number of rule set elements.*
- (G6) *Minimize the maintenance effort in a changing system.*

In Chapter 5 these informal goals are stated more precisely by using mathematical descriptions.

5. Quantifiable Metrics and Sets for Usable AC Rule Sets

In this chapter, we formalize the goals G1 to G6 and define the mathematical foundations of our approach. We first describe the building blocks that are needed to formalize ABAC, which is used as a reference system for further definitions. This formalization is based on the formalization seen in [Yuan and Tong, 2005a] but build to be minimal and focused on the goals. The formalization provides the sets, metrics and optimization criteria that are used to evaluate the usability of an access control rule set.

The following nomenclature is used in this section:

- $x \in X$ denotes that x is an element of set X .
- $|X|$ is the cardinality of set X (with a single exception in Section 5.4 for $|S_{G5}|$ where a different definition is used).
- $\mathcal{P}(X)$ is the powerset of set X .
- $X \times Y$ is the Cartesian product of set X and set Y .
- $X \subseteq Y$ denotes that set X is a subset of set Y .
- $X \supseteq Y$ denotes that set X is a superset of set Y .
- $X \cup Y$ is the union of set Y and set X .
- $X \triangle Y$ is the symmetric difference (XOR) of set X and set Y .
- $X \setminus Y$ is the relative complement of set Y in set X .

In addition the existential quantifier \exists and the logical conjunction \wedge are used.

5.1. Basic Building Blocks

The basis for the formalization is given with the following definitions. We follow the general set nomenclature, where capital letters refer to sets and non-capital letters to single elements. All sets are assumed to be finite.

Definition 1. *Entities.*

An entity is a subject, e.g., a person, that could be granted access to an object. The set of all entities is referred to as W (all possible entities, i.e., “the World”). The set E describes all entities in a system S , where $W \supseteq E$. The set B describes the set of owners of a system, where $W \supseteq B$.

Definition 2. *Attributes.*

Attributes are properties of entities such as ID number, age, gender, roles or security level. The set of all attributes is referred to as \mathring{A} and a subset of \mathring{A} is called A .

Definition 3. *Objects.*

Objects are anything that access rights can be assigned to, e.g., a file or directory. The set of all objects is referred to as O , the set of all objects in a system is called D (e.g., “Data” in form of all files of an information system), and the subsets of D are named H (e.g., a subdirectory or “hierarchy” in a Windows-based system). In short: $O \supseteq D \supseteq H$.

Definition 4. *Set of Access Decisions.*

There are two possible outcomes for an access request: allow or deny. We refer to the set of access decisions as $Z = \{-1, 1\}$, where -1 means DENY and 1 means ALLOW.

5.2. Derived Building Blocks

The following building blocks are constructed using the basic building blocks introduced above.

Definition 5. *Rule and Rule Set.*

A rule describes the relation between single attributes, objects, and access decisions and is written as a 3-tuple. For instance, the rule $(\{Students\}, \{Printer\}, 1)$ states that entities with the attribute *Students* are *allowed* to access the object *Printer*. A list R of n rules is called a rule set. We use the following notation: $R = (r_1, \dots, r_n)$, where $r_i = (A_i, H_i, z_i)$ and r_i refers to the i^{th} rule of the rule set R . H_i refers to subsets of D , and A_i refers to subsets of \mathring{A} .

Definition 6. *System.*

A system S is an environment described by a quadruple that consists of two sets of entities E and B , a set of objects D , and a set of access control rules R . It is defined as:

$$S = (B, E, D, R) \in \wp(W) \times \wp(W) \times \wp(O) \times (\wp(\mathring{A}) \times \wp(D) \times Z)^n,$$

where B denotes the set of owners of the system, i.e., the entities that define the access control rules for this system and $n = |R|$.

Further, we define two functions. Function f_A is used for extracting all attributes from an entity. Function $f_{request}$ provides access decisions.

Definition 7. *Attribute Extraction Function f_A .*

The attribute extraction function is defined as:

$$f_A : W \rightarrow \wp(\mathring{A}), w \mapsto f_A(w) := A_w.$$

It returns the set of attributes A_w that belong to entity w .

Definition 8. *Access Decision Function $f_{request}$.*

Let w be the requesting entity and H_w be the requested information i.e., a set of Objects. Let $z_{default}$ denote the default access decision in case that no rule is applicable to an access request, n denote the number of rules in R and $i \in \mathbb{N}$, $1 \leq i$. The access decision function $f_{request}$ is defined as follows:

$$f_{request}(w, H_w) := f_{req}(f_A(w), H_w, 1) \text{ or}$$

$$f_{request}(A_w, H_w) := f_{req}(A_w, H_w, 1), \text{ where}$$

$$f_{req} : (\wp(\dot{A}) \times \wp(D) \times \mathbb{N} \rightarrow Z, (A_w, H_w, i) \mapsto f_{req}(A_w, H_w, i)$$

$$f_{req}(A_w, H_w, i) := \begin{cases} z_{default} & \text{if } (i > n), \\ z & \text{if } A_w \supseteq A_i, H_w \subseteq H_i \\ & \quad \cdot (A_i, H_i, z_i) = r_i, \\ f_{req}(A_w, H_w, i+1) & \text{else.} \end{cases}$$

For most systems, one would typically use $z_{default} = -1$. n denotes the number of rules in the rule set.

Up to this point, we have provided a formalization of ABAC. In order to be able to evaluate whether a given rule set actually fits the system owners' intention, we provide a notation with regard to the intended behavior of the access control mechanism.

Definition 9. *Owners' Intention.*

The function $f_{intended}$ specifies the owners' intention with regard to access control decisions and is given as:

$$f_{intended} : (W \times \wp(D)) \rightarrow Z, (w, H) \mapsto f_{intended}(w, H)$$

$$f_{intended}(w, H) := \begin{cases} 1 & \text{if } b \in B \text{ wants } f_{request}(w, H) = 1, \\ -1 & \text{else.} \end{cases}$$

In practice, it can be challenging to acquire the function $f_{intended}$. A possible solution could be to observe system usage over a period of

time and use this information to approximate $f_{intended}$. More thoughts about the owners' intention can be found in Chapter 7.

5.3. Access Decision Sets

Before we concisely formalize the goals $G1$ to $G6$, we need to define eight access decision sets divided into two collections of four sets each. The first collection relates to the entities that are known as part of the system S , whereas the second collection relates to all entities that could be part of the system in the future which is required to address the problem of generalization of rule sets. In each collection two sets describe the owners intention and two how the system actually decides (for each one for allows and one for denies).

Definition 10. *System Access Decision Sets.*

The two sets $M_{E_{Allow}}$ and $M_{E_{Deny}}$ contain all entity, object tuples for which the access decision function $f_{request}$ would grant ($M_{E_{Allow}}$) or deny ($M_{E_{Deny}}$) access. The elements e are all elements of Set E and therefore part of the system.

The two sets $M_{E_{Wanted}}$ and $M_{E_{Unwanted}}$ contain all entity, object tuples for which the owner intends to grant ($M_{E_{Wanted}}$) or deny ($M_{E_{Unwanted}}$) access.

$$\begin{aligned} M_{E_{Allow}} &= \{(e, d) | f_{request}(f_A(e), \{d\}) = 1\}, \\ M_{E_{Deny}} &= \{(e, d) | f_{request}(f_A(e), \{d\}) = -1\}, \\ M_{E_{Wanted}} &= \{(e, d) | f_{intended}(f_A(e), \{d\}) = 1\}, \\ M_{E_{Unwanted}} &= \{(e, d) | f_{intended}(f_A(e), \{d\}) = -1\}. \end{aligned}$$

Definition 11. *World Access Decision Sets.*

The four sets described here are defined parallel to the sets in definition 10. The only difference is that now all entities are considered ($w \in W$).

$$\begin{aligned} M_{W_{Allow}} &= \{(w, d) | f_{request}(f_A(w), \{d\}) = 1\}, \\ M_{W_{Deny}} &= \{(w, d) | f_{request}(f_A(w), \{d\}) = -1\}, \\ M_{W_{Wanted}} &= \{(w, d) | f_{intended}(f_A(w), \{d\}) = 1\}, \\ M_{W_{Unwanted}} &= \{(w, d) | f_{intended}(f_A(w), \{d\}) = -1\}. \end{aligned}$$

5.4. Security and Usability Metrics

The Definitions 1 to 11 are used to formally define the sets S_{Gi} , where $1 \leq i \leq 6$. The sets S_{Gi} correspond to the security and usability metrics related to the goals Gi . The elements of a set S_{Gi} are the rules that contradict a goal Gi .

The criteria to achieve a goal Gi is therefore to minimize the number of elements in S_{Gi} : *minimize*($|S_{Gi}|$). The following definitions can be used to rate the usability of an access control rule set or to compare two different rule sets. The formalized definitions for S_{Gi} are:

(S_{G1}) *Cases where too much is allowed (allow not more than the owners want to be allowed):*

$$S_{G1} = M_{E_{Allow}} \setminus M_{E_{Wanted}}.$$

(S_{G2}) *Cases where too little is allowed (allow everything the owners want to be allowed):*

$$S_{G2} = M_{E_{Wanted}} \setminus M_{E_{Allow}}.$$

(S_{G3}) *Unnecessary rules (a rule must not be fully covered by another rule of the same rule set):*

$$S_{G3} = \{(r_i, r_j). 0 < i \leq n-1 \wedge i < j \leq n \\ \wedge A_j \supseteq A_i \wedge H_j \subseteq H_i \wedge z_j = z_i\}.$$

(S_{G4}) *Contradictory rules (two rules within the same rule set must not conflict):*

$$S_{G4} = \{(r_i, r_j). 0 < i \leq n-1 \wedge i < j \leq n \wedge \exists(e, h) \\ .(f_A(e) \supseteq A_i, h \subseteq H_i, z) \exists(f_A(e) \supseteq A_j, h \subseteq H_j, -z)\}.$$

With regard to S_{G4} , the default access decision $z_{default}$ is not considered a contradiction since it is not part of the rule set itself.

(S_{G5}) *Number of rules in the rule set (minimize the complexity of rules and rule set):*

$$S_{G5} = R, \quad \text{and}$$

$$|S_{G5}| := \sum_{(A_i, H_i, Z_i) \in R} |A_i| + |H_i| + 1.$$

It is important to notice that our definition here overrides the default cardinality operator.

(S_{G6}) *Cases that will lead to wrong access decisions in the future (minimize the maintenance effort in a changing system):*

$$S_{G6} = M_{W_{Allow}} \triangle M_{W_{Wanted}} \cup M_{W_{Deny}} \triangle M_{W_{Unwanted}}.$$

In practice it is very difficult to build the set S_{G6} , since it takes into account a future state as it considers entities that are not yet part of the system, but are going to join it at a future time. The fields of knowledge engineering and machine learning refer to this problem as the generalization or overfitting problem [Mitchell, 1997]. The interviews with

IT support professionals in the pilot study (and informal discussions with scientists from the knowledge engineering field) indicate that an optimized $|S_{G3}|$, $|S_{G4}|$, and $|S_{G5}|$ would have a positive effect on $|S_{G6}|$ (e.g., minimizing the rule set leads to more general rules that are less overfitting).

5.5. The Cost of Wrong Access Decisions

The two types of failures related to access control decisions are: decisions that should have been denied but were not, i.e., the elements in S_{G1} , or decisions that should have been allowed but were not, i.e., the elements in S_{G2} . Naturally, the consequences of failures vary e.g., granting access to a confidential file carries a higher cost than granting access to a non-critical system file. To capture such distinctions between different failures regarding their impact on the system or its users, the function

$$f_{S_{G1}} : D \rightarrow \mathbb{R}, d \mapsto f_{S_{G1}}(d)$$

is used to get the cost of an incorrect allow and

$$f_{S_{G2}} : D \rightarrow \mathbb{R}, d \mapsto f_{S_{G2}}(d)$$

is used to get the cost of an incorrect deny.

The value $cost_{S_{G1}}$, which is related to S_{G1} (incorrect allows) and attributed to an access control rule set, is

$$cost_{S_{G1}} = \sum_{d \in X} f_{S_{G1}}(d),$$

where $X = \{d \mid (e, d) \in M_{E_{Allow}} \setminus M_{E_{Wanted}}\}$.

And the value $cost_{S_{G2}}$, which is related to S_{G2} (incorrect denies) and attributed to an access control rule set, is

$$cost_{S_{G2}} = \sum_{d \in Y} f_{S_{G2}}(d),$$

Table 5.1.: Security and Usability Metrics: S_{Gi}

S_{G1}	$= M_{E_{Allow}} \setminus M_{E_{Wanted}}$
S_{G2}	$= M_{E_{Wanted}} \setminus M_{E_{Allow}}$
S_{G3}	$= \{(r_i, r_j). 0 < i \leq n-1 \wedge i < j \leq n$ $\wedge A_j \supseteq A_i \wedge H_j \subseteq H_i \wedge z_j = z_i\}$
S_{G4}	$= \{(r_i, r_j). 0 < i \leq n-1 \wedge i < j \leq n \wedge \exists(e, h)$ $\cdot (f_A(e) \supseteq A_i, h \subseteq H_i, z) \nexists (f_A(e) \supseteq A_j, h \subseteq H_j, -z)\}$
S_{G5}	$= R, \text{ and } S_{G5} := \sum_{(A_i, H_i, Z_i) \in R} A_i + H_i + 1$
S_{G6}	$= M_{W_{Allow}} \triangle M_{W_{Wanted}} \cup M_{W_{Deny}} \triangle M_{W_{Unwanted}}$

where $Y = \{d | (e, d) \in M_{E_{Wanted}} \setminus M_{E_{Allow}}\}$.

Finally, we define $total\ cost = cost_{S_{G1}} + cost_{S_{G2}}$.

5.6. Summary

In this chapter the requirements presented in chapter 4 are formalized into quantifiable metrics and sets S_{Gi} as seen in Table 5.1. A metric for each requirements is defined as the cardinality of the sets $|S_{Gi}|$. $|S_{G5}|$ has a slightly adjusted definition to optimize the explanatory power of the metric. The sets $|S_{Gi}|$ have to be minimized to optimize security and usability of the corresponding access control rule set.

The next chapter presents the evaluation of the formalization, which is achieved with the help of two user studies.

6. Evaluation of Formal Metrics and Sets

In this chapter we evaluate the formalization presented in chapter 5. Three hypotheses are tested:

Hypothesis H1: The sets and metrics help users produce better rule sets.

Hypothesis H2: The usability scores computed using the metrics in Chapter 5 correspond to rankings obtained from IT support professionals when evaluating the translation of policies into access control rule sets (related to $G1$ and $G2$).

Hypothesis H3: The usability scores computed using the metrics in Chapter 5 correspond to rankings obtained from IT support professionals when evaluating the understandability and manageability of access control rule sets (related to $G3$, $G4$ and $G5$).

These hypotheses are tested with the help of two user studies.

In User Study 1 (see Section 6.2.1), participants were asked to complete a computer-assisted task regarding the optimization of an access control rule set. Two test conditions were used for completing the task: *without the sets and metrics (WOS)* and *with support of the sets and metrics (WS)*. Our results show that users implement significantly better rule sets when supported by tools that provide the measurements S_{Gi} and $|S_{Gi}|$ from chapter 5.

In User Study 2 (see Section 6.2.2), the participants were IT support professionals. They were asked to evaluate and rank the rule sets that were obtained from User Study 1 based on their own experience and knowledge.

Two evaluation criteria were defined:

- (a) *how accurately the rule sets implement the access control policy and*
- (b) *how easily the rule sets can be understood and managed.*

Afterward, the results from the experts are compared to rankings obtained by using the provided metrics $|S_{Gi}|$. A high correlation is found between the rankings obtained through using the metrics and the opinions of the experts. These results validate *Hypothesis H2* and *Hypothesis H3*, therefore validating *Hypothesis H1*.

Section 6.1 is used as an example how to use the metrics and sets to optimize an access control rule set and as a scenario for User Study 1 in Section 6.2. Limitations of the formalization and of the user studies are discussed in Section 6.3. Section 6.4 discusses open questions and shows possibilities for future refinements.

6.1. Example

In this section we provide a scenario to illustrate how the security and usability metrics presented in the previous section can be used to measure, compare and optimize rule sets in order to construct usable access control rule sets, i.e., rule sets that are easy to understand and manage and that reflect the desired access control policy. The scenario presented in this section is the same as the one used in User Study 1, presented in the next section. The scenario is described by:

- a table of entities and their attributes,
- a table with the description of a file system and the desired policy,
- a graphical representation of the same file system and the desired policy,
- two tables describing access control rule sets.

Table 6.1.: Entity–Attribute–Relationship Table. The ‘x’ marks indicate that a given attribute (column) is associated with a given entity (row), e.g., entity 1 has attributes A3, A4 and A7.

Entity	Attributes					
	A3	A4	A5	A6	A7	A8
1	x	x			x	
2	x	x			x	
3			x	x	x	
4	x		x		x	
5	x	x	x		x	
6		x	x		x	
7	x		x		x	
8			x		x	x

In the scenario, each entity has an arbitrary number of attributes assigned to it. There are eight entities (1 to 8) and six attributes (A3 to A8) and 12 Files (a.txt to l.txt). Table 6.1 illustrates the relationship between entities and attributes.

The scenario describes a file system. It defines which files an entity should or should not have access to. The file system mimics a MS-Windows© file system with ‘C:’ as its root. The directories are assigned the letters ‘a’, ‘b’, and ‘c’. All files have a ‘.txt’ extension. The file system is presented in Table 6.2. Table 6.2 also includes the $cost_{S_{G1}}$ associated with each file.

A graphical representation of the file system is shown in Figure 6.1.1. It also depicts the $cost_{S_{G1}}$ for files ‘d.txt’, ‘f.txt’ and ‘j.txt’. The $cost_{S_{G1}}$ of each file (except ‘d.txt’, ‘f.txt’ and ‘j.txt’) is 10 points and the $cost_{S_{G2}}$ of each file is 5.

Tables 6.3 and 6.5 present a rule set each. The rule sets are two different implementations of the access control policy represented in the entity-attribute relationship shown in Table 6.1, applied to the file system described in Table 6.2.

The compilation of the scores $|S_{Gi}|$, $cost_{S_{G1}}$ and $cost_{S_{G2}}$ (associated with S_{G1} and S_{G2} respectively) in Tables 6.4 and 6.5 represent the results

Table 6.2.: A description of the file system showing which files an entity should have access to and the $cost_{S_{G1}}$, if an unauthorized access is allowed, assigned to each file. The $cost_{S_{G2}}$, if an authorized access is denied, assigned to each file, is 5 for each file.

$cost_{S_{G1}}$	File Name	Entities that should have access
10	C:\a\ a.txt	1
10	C:\b\ b.txt	1, 2, 3, 4, 5, 6, 7, 8
10	C:\c\ a\ c.txt	3, 5, 6
50	C:\c\ a\ b\ d.txt	3
10	C:\c\ a\ c\ e.txt	3
80	C:\c\ b\ a\ f.txt	1, 2, 3, 4, 5, 7
10	C:\c\ b\ b\ g.txt	1, 2, 3, 4, 5, 7
10	C:\c\ b\ c\ h.txt	1, 2, 3, 4, 5, 7
10	C:\c\ c\ a\ i.txt	3, 8
1	C:\c\ c\ b\ j.txt	1, 2, 3, 8
10	C:\c\ c\ c\ a\ k.txt	3, 8
10	C:\c\ c\ c\ b\ a\ l.txt	3, 8

obtained from each rule set and take into account the file system and the desired entity-attribute relationship of the scenario.

It can be difficult to analyze the two rule sets and decide which better fits the scenario without considering the scores $|S_{Gi}|$.

With the $|S_{Gi}|$ scores, it is much easier to compare both rule sets, as the scores provide a indication of the quality of each rule set with regard to the defined goals for security and usability. The values of $cost_{S_{G1}}$ and $cost_{S_{G2}}$ are the most important values to compare when looking

Table 6.3.: Access Control Rule Set One

#	Path	Attributes	Decision
1	c:\a\	A6	DENY
2	c:\	A6	ALLOW
3	c:\b\	A7	ALLOW
4	c:\c\ a\	A4, A5	ALLOW
5	c:\c\ b\	A7	ALLOW
6	c:\c\ c\	A8	ALLOW
7	c:\c\ c\ c\	A8	ALLOW

Table 6.4.: Usability scores computed using the metrics in Chapter 5 of Access Control Rule Set One

Goal	$ S_{Gi} $	$cost_{S_{Gi}}$
<i>G1 (Too much allowed)</i>	10	320
<i>G2 (Too little allowed)</i>	3	15
<i>G3 (Unnecessary rules)</i>	1	-
<i>G4 (Contradicting rules)</i>	1	-
<i>G5 (Elements in rule set)</i>	22	-

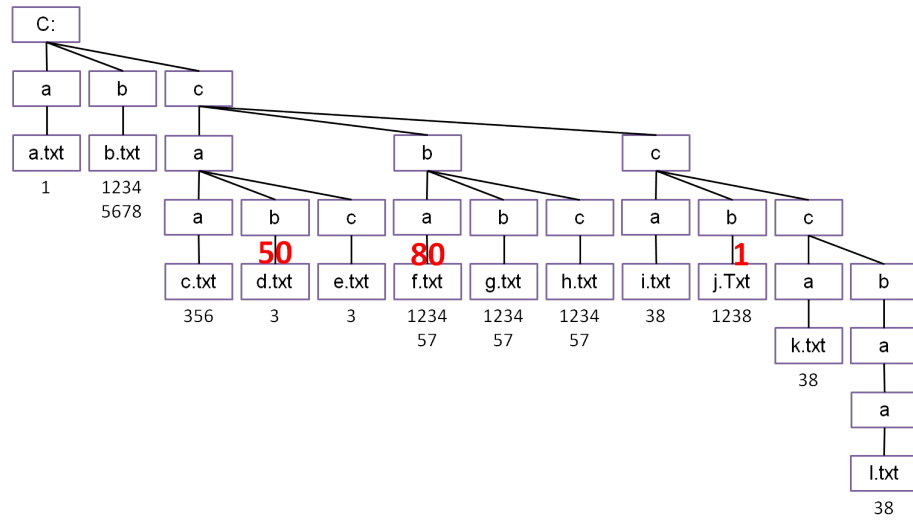


Figure 6.1.1.: A graphical representation of the file system. It shows the files an entity should have access to (below the file name) and the non-default values $cost_{S_{G1}}$ (in red). The default value is 10.

at the accuracy of the rule sets, i.e., how accurate they are in making correct access control decisions. $|S_{G3}|$, $|S_{G4}|$, and $|S_{G5}|$ are related to the manageability of the access control rule set.

Regarding accuracy, rule set number two (Table 6.5) is superior to rule set number one (Table 6.3), as it has better (lower) scores for $|S_{G1}|$, $|S_{G2}|$, $cost_{S_{G1}}$ and $cost_{S_{G2}}$, as shown in Tables 6.4 and 6.6. Rule Set One has a better score than Rule Set Two only for $|S_{G5}|$, having three less elements. In all other categories, Rule Set Two is superior or equal to Rule Set One.

Table 6.5.: Access Control Rule Set Two

#	Path	Attributes	Decision
1	c:\b\	A7	ALLOW
2	c:\c\	A6	ALLOW
3	c:\c\ a\ a\	A4, A5	ALLOW
4	c:\c\ b\	A3	ALLOW
5	c:\c\ c\	A8	ALLOW
6	c:\c\ c\ b\	A3, A4, A5	DENY
7	c:\c\ c\ b\	A3, A4	ALLOW

Table 6.6.: Usability scores computed using the metrics in Chapter 5 of Access Control Rule Set Two

Goal	$ S_{Gi} $	$cost_{S_{Gi}}$
$G1$	0	0
$G2$	1	5
$G3$	0	-
$G4$	1	-
$G5$	25	-

When optimizing a rule set, the metrics add additional information. For instance, regarding Rule Set One, the metric S_{G1} indicates 10 errors with an associated $cost_{S_{G1}}$ of 320. The $cost_{S_{G2}}$ of denying authorized accesses is 15 (in row G2). Inspecting the 10 elements in set S_{G1} , the following information can be extracted: $\{(E5, d.txt), (E5, e.txt), (E6, d.txt), (E6, e.txt), (E6, f.txt), (E6, g.txt), (E6, h.txt), (E8, f.txt), (E8, g.txt), (E8, h.txt)\}$. The cost function $f_{S_{G1}}$ shows that $(E6, f.txt)$ and $(E8, f.txt)$ both have cost values of 80 and are the most critical errors, i.e., the errors with the highest possible cost. These two errors can be eliminated by changing the attribute A7 to A3 in rule #5. Then, by recalculating the results we obtain: $|S_{G1}| = 4$, $|S_{G2}| = 3$, $|S_{G3}| = 1$, $|S_{G4}| = 1$, $|S_{G5}| = 22$ with a $cost_{S_{G1}}$ of 120 and a $cost_{S_{G2}}$ of 15. These values indicate a significant improvement over the previous version of the rule set.

6.2. Validation of the Formal AC Rule Set Definitions

We validated the sets and metrics and their usefulness in supporting the users in optimizing of access control rule sets by testing three hypotheses:

Hypothesis H1: The sets and metrics help users produce better rule sets.

Hypothesis H2: The usability scores computed using the metrics in Chapter 5 correspond to rankings obtained from IT support professionals when evaluating the translation of policies into access control rule sets (related to *G1* and *G2*).

Hypothesis H3: The usability scores computed using the metrics in Chapter 5 correspond to rankings obtained from IT support professionals when evaluating the understandability and manageability of access control rule sets (related to *G3*, *G4* and *G5*).

We tested these hypotheses with the help of two user studies. User Study 1 was aimed at gathering data from both non-experts and IT support professionals regarding the creation of rule sets that match the system owners' intention, with and without the support of our proposed sets and metrics. The outcome from User Study 1 was used as input to User Study 2 where expert ratings regarding the usability of these rule sets were compared to the values obtained through our metrics. The output of the user studies was analyzed in Section 6.2.3 and the limitations of our user studies are listed and discussed in Section 6.3.

6.2.1. User Study 1 - Optimizing Rule Sets

In User Study 1, participants were asked to complete a computer-assisted task regarding the optimization of an access control rule set. Two test conditions were used for completing the task: *without the sets and metrics* (WOS) and *with support of the sets and metrics* (WS).

Method Twelve participants took part in the study. The participants were recruited by sending e-mails to mailing lists and asking for participation in an IT research user study. Two thirds were non-experts regarding access control configuration and management. The other four participants were IT support professionals, who manage access control mechanisms on a regular basis. One of the IT support professionals had also participated in the pilot study. The age of the participants ranged between twenty and fifty-five ($\mu = 34.5$, $\sigma = 8.1$) and four participants were female. Seven of the participants were graduate students, one had a PhD degree, three held degrees from universities of applied sciences, and one had no university degree. No financial incentive was offered to the participants for taking part in the study.

A between subjects design was applied in this user study. The study was designed as a laboratory experiment. The experiment was individual, i.e., one participant at a time. Participants had the task explained by a supervisor (the task was described in writing and handed out at the beginning of the experiment). The supervisor answered questions regarding the task description, informed the participants about the maximum time allowed and enforced this time limit. The time allowed was 20 minutes (plus the time required to explain the task). Participants were encouraged to vocalize their line of thought.

The task was to, given an existing access control rule set, minimize the cost associated with it by changing, adding or deleting rules. The rule set was given to the participants in the form of an MS Excel spreadsheet to minimize possible bias due to the presentation of the task, as all participants were familiar with this spreadsheet application.

There were two conditions used in the laboratory experiment: *without support of the sets and metrics (WOS)* and *with support of the sets and metrics (WS)*. The IT support professionals were equally distributed between the two conditions to avoid impact of their expertise on the results. The rest of the participants were randomly assigned to one of the test conditions.

In *WOS*, participants were asked to optimize the rule set without additional support by any sets and metrics (apart from the spreadsheet application). In *WS*, the spreadsheet application was programmed to return all sets and metrics provided by our formalization, including the *total cost* ($= cost_{S_{G1}} + cost_{S_{G2}}$) associated with the rule set, which was displayed when the participant clicked a button labeled UPDATE in the spreadsheet application interface.

The participants were informed what rule sets are, how rules are expressed (in terms of ALLOW/DENY decisions), and how they are processed (from top to bottom). In particular, the participants were informed about the following: DENY rules have precedence over ALLOW rules (this is done by moving all DENY rules to the beginning of the rule set to mimic the behavior of a windows file-system), there is a default implicit DENY ALL rule at the end of the rule set, and if a rule applies to a directory then all its sub-directories and files inherit that same rule.

The task description contained: Table 6.1; Table 6.2 and its graphical representation (Figure 6.1.1); and Table 6.3, which presented the initial rule set that had to be modified by the participant to adhere to the desired policy.

At the end of the experiment, participants handed in the access control rule sets that they produced. Twelve rule sets were obtained. Participants in the condition *WOS* were asked, after handing in their rule sets, to repeat the experiment with the support of the sets and metrics, i.e., following the *WS* test condition, and produced six new sets of rules. The six additional rule sets were used to increase the size of the input to User Study 2 and used only to test *Hypothesis H2* and *Hypothesis H3*. Naturally, the additional rule sets were not used to test *Hypothesis H1* as they were affected by experiment order and learning effects. Order and learning effects that affect the additional rule sets are not relevant to the objectives of User Study 2.

Acquired Data The outcome of User Study 1 is three times six access control rule sets (six from test condition *WOS*, six from test condi-

Table 6.7.: Spearman's rank correlation test between the automatically produced rankings and the rankings obtained in User Study 2. *Proposal* refers to the automatically produced ranking and *Result 1* to *Result 4* to the results obtained from IT support professionals. $N = 18$ for all cases. A correlation coefficient of 1.000 would represent perfect correlation to the proposal. A correlation coefficient of 0.000 would represent no correlation at all.

Spearman's rho Correlation Coefficients					
	Proposal	Result 1	Result 2	Result 3	Result 4
<i>Hypothesis 2</i>	1.000	.908**	.967**	.971**	.955**
<i>Hypothesis 3</i>	1.000	.922**	.820**	.874**	.777**

tion *WS*, and the six additional ones). These rule sets were used as input for User Study 2.

6.2.2. User Study 2 - Approach Versus Experts

In User Study 2, the participants were IT support professionals. They were asked to evaluate and rank the rule sets that were obtained from User Study 1 based on their own experience and knowledge.

Two evaluation criteria were defined:

- (a) *how accurately the rule sets implement the access control policy and*
- (b) *how easily the rule sets can be understood and managed.*

Methodology The 18 rule sets generated in User Study 1 were tested in 8 sub-experiments, that were performed with $N = 18$ each. Four IT support professionals took part in the evaluation according to criterion (a) and four took part in the evaluation according to criterion (b). Each expert processed all 18 rule sets. The IT support professionals were recruited from business and public sectors (universities). One of the participants had taken part in the pilot study and User Study 1. Two of the IT support professionals had taken part in the pilot study but not in User Study 1. The IT support professionals were not informed about

the metrics or goals of the user study. All of them managed access control mechanisms on a regular basis and had worked several years in positions related to IT support. Again, no financial incentive was offered to the participants¹.

The collection of access control rule sets was sent to the IT support professionals by electronic mail. The ordering of the rule sets was randomized before being sent to the participants. The participants were asked to order the rule sets regarding criteria (a) and (b) and provide a short description of their approach for evaluating the rule sets. No time limit was set to complete the ranking.

Acquired Data The results of User Study 2 are two rankings from each expert. One reflects the opinion of the IT support professionals regarding how accurately the rule sets implement the access control policy and the other reflects how easily, in their opinion, the rule sets can be understood and managed. The participants claimed to have taken up to seven hours to complete the task and one stated that the analysis of some rule sets took close to one hour to analyze. The IT support professionals reported different approaches and methods used in their rankings. The main aspects reported when evaluating manageability of rule set were the following: the time needed to read and understand it, the number of elements in it, and the number of DENY rules. The translation of the defined policy into a rule set was evaluated according to the number of security gaps and wrongly denied accesses. Next, each outcome of the sub-experiments of User Study 2 was tested for correlation with the outcome obtained using our sets and metrics.

6.2.3. Results and Evaluation

In this section we describe how we validated our three hypotheses. Hypotheses *H2* and *H3* were validated by the strong correlation between the ranking produced by IT support professionals and the rank-

¹We again promised to inform them first-hand about our findings and conclusions.

Table 6.8.: Independent Samples t-test

Levene's Test for Equality of Variances			t-test for Equality of Means						
Equal variances	F	Sig.	t	df	(2-tailed) Sig.	Mean Diff.	Std. Error Diff.	95% Confidence Interval of the Diff.	
								Lower	Upper
assumed	3.005	.114	3.692	10	.004	153	41.446	60.653	245.347
not assumed			3.692	7.621	.007	153	41.446	56.592	249.408

ing obtained by using our usability scores computed using the metrics in Chapter 5. After validating *Hypothesis H2*, we validated *Hypothesis H1*.

To validate *Hypothesis H2* and *Hypothesis H3*, we compared the rankings produced by the IT support professionals in User Study 2 and the rankings generated using our usability scores computed using the metrics in Chapter 5. For testing *Hypothesis H2*, we compared the list of the four rankings produced using criterion (a) and the rankings generated using the *total cost* metric ($cost_{S_{G1}} + cost_{S_{G2}}$). *Hypothesis H3* was tested by comparing the list of the four rankings produced using criterion (b) and the rankings generated using S_{G3} , S_{G4} and S_{G5} .

Spearman rank correlation coefficients were computed to assess the relationship between the rankings. Overall, there was a significant positive correlation between the automatically produced rankings and the rankings obtained from User Study 2, as shown in Table 6.7. The results from these tests validate both *Hypothesis H2* and *Hypothesis H3*.

The correlation was higher for *Hypothesis H2* than for *Hypothesis H3*. This was expected because there is a common methodology to evaluate how accurately a rule set implements an access control policy by analyzing the results for security gaps and non-granted legitimate access rights. The IT support professionals used similar methodologies to rank the rule sets according to criterion (a). Interestingly, all IT support professionals made small mistakes by overlooking some gaps. However, when ranking the rule sets according to their manageability, the IT support professionals used a wider variety of approaches, such as counting

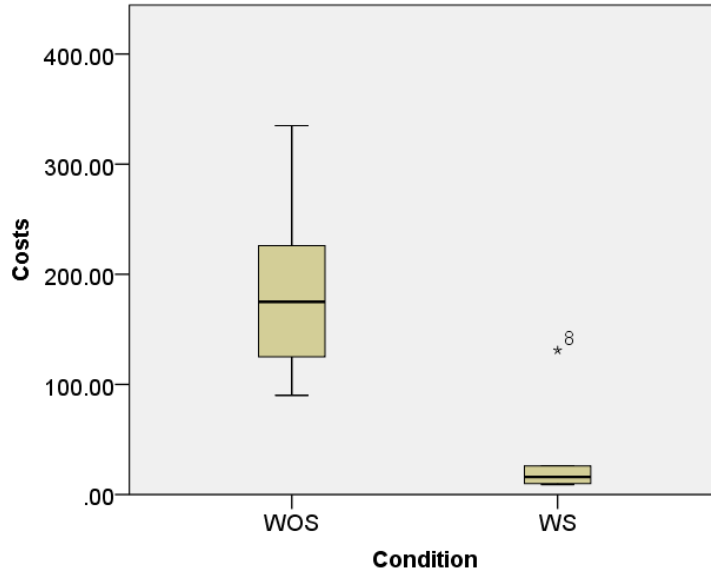


Figure 6.2.1.: Box plot showing the results of User Study 1. They are presented with 0.95 confidence interval. The WS group (with support of our metrics and sets) performed significantly better (lower values) regarding the cost score than the WOS group (without support of our metrics and sets).

the number of DENY rules, the time spent to understand the rule set, or deciding intuitively.

User Study 2 aimed to evaluate whether the values $|S_{G3}|$, $|S_{G4}|$ and $|S_{G5}|$ can be used to provide results that are similar to results obtained from IT support professionals. The results from User Study 2 showed a strong correlation between the results obtained from the IT support professionals and the results that were automatically generated by software that implements our proposed formalization. This result validated the expressiveness of $|S_{G3}|$, $|S_{G4}|$ and $|S_{G5}|$.

After validating *Hypothesis H2*, we were able to test *Hypothesis H1* by calculating the *total cost* metric of each access control rule set produced in the User Study 1 and comparing the results from the WOS and WS groups.

The Box plot in Figure 6.2.1 summarizes the results obtained from User Study 1. The mean total cost for condition WOS (no support)

was significantly higher ($\mu = 187.7$, $\sigma = 36.7$) than the total cost for condition *WS* (with support) ($\mu = 34.7$, $\sigma = 19.5$). This difference in the results is also shown in Table 6.8, which compares the results for the two conditions using independent samples t-test for the test conditions *WOS* ($\mu = 187.7$, $\sigma = 36.7$) and *WS* ($\mu = 34.7$, $\sigma = 19.5$) for $t(3.692) = 7.621$ and $p = 0.007$.

The participants in *WS* performed significantly better than the participants in *WOS*. The analysis of the results obtained from User Study 1 validated *Hypothesis H1* by showing that our sets and metrics help users to produce significantly better rule sets.

6.3. Remarks

A real case regarding the management of an access control rule set can easily involve tens of thousands of objects and as many entities. Still, we deliberately designed User Study 1 with few objects (12) and entities (8). Our decision to limit the number of objects and entities was based on two considerations. First, a more complex scenario would be more difficult for participants to understand given the conditions and practical limitations of the study. Second, User Study 1 is close to a worst case scenario with respect to the performance of our approach as a more complex scenario would also increase the space for misconfiguration and errors. As our metrics are designed to allow identification of such cases, it is expected to produce significantly better results in a more complex and non-controlled environment.

The sample size of User Study 1 (twelve participants) is not large, but enough to obtain significant results from the statistical tests on the collected data. In User Study 2, four IT support professionals ranked the 18 rule sets produced in User Study 1. Increasing the number of participants in User Study 1 would result in a large sample of rule sets and it would also increase the number of rule sets each IT professional would need to rank. A practical limitation of our study is that all the

participants were volunteers, and the amount of effort required from the experts was considerable. The four IT professionals in User Study 2 produced similar rankings, which suggests that four was sufficient for our evaluation. The IT professional volunteers were very positive about our studies and, following User Study 1, two of them independently asked the study supervisor about the possibility of integrating our tools into their workflow, as they strongly believed that the tools would facilitate their work.

A limitation of User Study 2 is that it cannot individually validate the metrics $|S_{G3}|$, $|S_{G4}|$ and $|S_{G5}|$, but rather only the composition of all factors together. Hence, we were not able to evaluate the impact of each individual metric when testing rule sets for their manageability. It would be interesting to analyze the individual impact of each metric to obtain even better results.

6.4. Discussion

In this section we discuss our findings, open challenges towards introducing new factors in our metrics and opportunities for future work.

The six goals for building usable access control rule sets presented in our work were derived from the pilot study. The goals formalize the metrics used by experts to evaluate rule sets. This set of goals is not comprehensive and is a subset of goals for building usable access control rule sets. Other metrics could be included in the set if they are found relevant in future studies. For instance, the design of the user interface was never mentioned during the interviews of the pilot study, but it may be an important aspect for most users. Another factor that is not reflected by our metrics is the indirect interdependency of rules, which may impact the usability of rule sets. Extending the set of metrics could lead to better rule sets, but determining their importance would require further testing and evaluation.

A challenging aspect of our building blocks presented in Section 9 is the formalization of the owners' intention, $f_{intended}$. Obtaining the owners' intention is out of the scope of this dissertation but it is a key aspect to be considered in future work. Solutions would possibly involve direct interaction with the owner using tools, such as a reactive access control mechanism [Mazurek et al., 2011], psychological testing, questionnaires or observation of the owners' behavior in using and sharing data.

Another important aspect to be carefully analyzed is the use of cost functions. Assignment of costs is highly subjective and dependent on the nature of data. Costs are relevant for defining levels of importance for different objects (i.e., different objects with different costs) and goals (i.e., different costs for too much allowed access and too little allowed access). Nevertheless, the metrics presented in this dissertation are independent of cost assignment. An interesting extension of this work would be to introduce cost functions for the sets S_{G3} , S_{G4} , S_{G5} and S_{G6} . The additional cost functions would be an important step towards building a single metric instead of multiple metrics to rate a rule set.

Optimizing a criterion could affect other criteria, therefore it is important to evaluate dependencies between criteria in future work. For instance, eliminating contradictions ($G4$) can sometimes lead to a more complex rule set ($G5$) as shown in the following example:

RULE 1: Alice is denied access to file.

RULE 2: Everyone is allowed access to file.

Above we have a short rule set with one contradiction. A non-contradicting rule set that describes the same scenario could be implemented as following:

RULE 1: Bob is allowed access to file.

RULE 2: Chris is allowed access to file.

RULE 3: Dave is allowed access to file.

...

RULE 23: Xena is allowed access to file.

RULE 24: Yuri is allowed access to file.

RULE 25: Zara is allowed access to file.

This rule set has more elements and no contradictions. Cost functions of the sets S_{G3} to S_{G6} would show the tradeoffs between multiple criteria.

6.5. Summary

In this chapter two user studies are described. The results validated the following three hypothesis.

Hypothesis H1: The sets and metrics help users produce better rule sets.

Hypothesis H2: The usability scores computed using the metrics in Chapter 5 correspond to rankings obtained from IT support professionals when evaluating the translation of policies into access control rule sets (related to $G1$ and $G2$).

Hypothesis H3: The usability scores computed using the metrics in Chapter 5 correspond to rankings obtained from IT support professionals when evaluating the understandability and manageability of access control rule sets (related to $G3$, $G4$ and $G5$).

The next chapter gives an outlook to future research and a conclusion to this dissertation.

7. Outlook and Conclusion

The generation of accurate and optimized access control rule sets is a complex task. It can be effectively approached by dividing the task into smaller sub-problems. The identified sub-problems are:

1. Data Accumulation
2. Rule Extraction and Optimization
3. Rule Set Inspection

To tackle these sub-problems, the results of this dissertation have to be integrated with research from the fields of psychology, machine learning, and human-computer interface to obtain a holistic approach.

7.1. Data Accumulation / Getting Owners' Intention

The first and maybe most important step is to acquire an initial access control policy. The process of getting initial access control information is often about extracting owners' intention into a machine readable form. This process can be problematic since users are not always capable of precisely and unambiguously defining their wishes.

Two approaches are conceivable. First, the classical approach, whereby users e.g., manually implement access control rights in an ABAC mechanism. This process makes a potentially large amount of errors possible. With the help of intuitively usable human-computer interfaces, questionnaires, and the metrics provided in this dissertation (see Chapter 5 and Chapter 6), the process can be made much easier for users.

Second, observations of behavior on a running system can be stored and interpreted as an initial access control policy. This is adequate as long as the observed entities behave correctly. To guarantee that no malicious behavior is integrated in the access policy, system owners have to check the gathered information afterward. This information may help owners in generating appropriate access rules.

7.2. Rule Extraction and Optimization

After the Data Accumulation step the access control information or initial rule set may be unrefined. Especially casual users may encounter difficulties. Additional tools like the metrics provided in Chapter 5 can help users to manually optimize access control rule sets as seen in Chapter 6.

The provided metrics of Chapter 5 also have the power to be used as optimization criteria for automated rule generation and optimization. The goal of a rule optimization algorithm is to minimize the sets S_{Gi} described in Section 5.4. The sets can be assigned to three different classes:

- Functional Sets (S_{G1} & S_{G2})
- Configuration Sets (S_{G3} & S_{G4} & S_{G5})
- Future Set (S_{G6})

It is difficult to optimize the Functional Sets and the Future Set directly since external information is needed:

To minimize the Functional Sets directly, the intention of the owners needs to be acquired to eliminate deviance to the actual system behavior as described in Section 7.1.

For the direct optimization of the Future Set S_{G6} , knowledge about the future state of the system would be required. This information is generally only available in part, if at all. Fortunately, S_{G6} improves when minimizing the configuration sets (see Section 5.4). A more readable

and easier to configure access control rule set also helps users find and correct errors regarding S_{G1} & S_{G2} . So by directly minimizing the Configuration Sets, indirect benefits for all the other sets are obtained.

However, as already mentioned, user preferences make full automation difficult. The next section gives a possible approach to minimize the required user interaction even further.

7.3. Rule Set Inspection

The majority of users nowadays are unable to properly configure security mechanisms, mostly because the mechanisms are not usable for them, as demonstrated in related work (see Section 2.6). To attain the goal of having usable security mechanisms, the best solution is to minimize the amount of user interactions and simplify configuration tasks. Automation is an appropriate solution for minimizing the amount of user interaction.

Fully automated access control policy generation is currently not possible and may never be because individual preferences must be taken into account (see Section 2.3), therefore generation still requires at least some user interaction. To address this problem, we propose a mechanism that helps users inspect rule sets and helps them reflect their individual preferences. We call this mechanism Interactive Rule Learning for Access Control. Interactive Rule Learning is designed to generate concise rule sets for Attribute-Based Access Control (ABAC).

7.4. Future Vision: Interactive Rule Learning

One important question to answer is: what is private and confidential data? There is no universal answer for this question because it depends on the person being asked. For that reason, it is important that the user's

preferences are taken into account in the decision of what is made public and what is kept secret. However, the users can not be expected to be satisfied with having to manually define all the rules for each device [Cranor and Garfinkel, 2005]. There is a need for some kind of automation that lets every user only select their preferences once or a small number of times, and subsequently have these preferences used automatically. In addition, a set of default rules suitable for most use cases enhances the usability of the security mechanism.

That way only slight user intervention is required to generate and maintain access rules. However, there still may remain a lot of interaction requests if the rules are too fine-grained or if the rules need to change often. Appropriate rules should not only cover the current context, but also future situations such as “any employee may access the file server”. When new employees are hired, they should immediately have the right to access the file server. An overly specific rule that does not generalize would be: “Mr. Jonson is allowed to access the printer”. Rules of this kind would be necessary for each employee, leading to a barely manageable set of rules. Excessively general rules are problematic, too, because they may allow unauthorized entities to perform restricted tasks. If e.g., a rule says “everyone is allowed to do anything”, the AC itself is pointless.

7.4.1. Example

In our example, we consider a family of four. Alice (a), Bob (b), and their children Charlie (c), a 17-year old, and Denise (d), an 8-year old. The set with elements $\{a, b, c, d\}$ is the family, and the subset with elements $\{a, b\}$ are the parents. In the family’s kitchen there are 3 new smart products: a smart coffee machine (x), a smart blender (y), and a smart oven (z).

We assume that newly bought devices come with a default set of access rules, which are defined by the smart product manufacturers. Since the manufactures cannot predict in which way the smart products are

going to be used, the factory settings for access control rules are fairly general. They follow the usage rules of similar non-smart products, i.e., everyone that physically interacts with a device is allowed to use it up to its full functionality. For instance, everyone locally interacting with a coffee machine is allowed to brew coffee.

Full control of a smart product is given to the first user who activates it. A smart product might be remotely controlled by its users (through smart devices) after it has been integrated into the home environment. *a* wants to configure and generates access rules for the 3 newly bought smart products (*x*, *y*, *z*), so that her family can best benefit from them. Three classes of access rights are preset in smart devices (those classes can later be reconfigured or changed):

1. *Full access*: the right to locally or remotely access a smart product and to manage its access rights.
2. *Remote and local access*: the right to locally and remotely access a smart product.
3. *Local access*: the right to locally access the smart products.

a wants to grant *b* with full control over all the smart products. *c* shall get access to the full functionality (locally and remotely), but shall not have administrative rights over the smart products. *d* shall not have any access to the devices, even by local interaction. Since the family often has guests, *a* wants them to be able to locally interact with the smart products, just as in a non-smart kitchen. The initial manually generated rule set has some errors (see Table 7.1):

Table 7.1.: Interactive Rule Learning Rule Set 1

1:	If (owner)	→	full access
2:	If (any)	→	local access
3:	If (a)	→	full access
4:	If (b)	→	full access
5:	If (c)	→	remote and local access
6:	If (d)	→	no access to x
7:	If (d)	→	no access to y
8:	If (a)	→	no access to z
9:	If (guest)	→	local access

- The first two rules are leftovers from the preloaded default rule set. The fact that *a* ignored them leads to two implications regarding requirements G3 (Make sure that a rule is not fully covered by another rule of the same rule set) and G4 (Two rules within the same rule set must not conflict). Rule 2 is a superset of rule 9, and it also contradicts rules 6, 7, and 8. Moreover, since some rules are redundant, their number is surely not minimal, which contradicts G5.
- Rule 9 is misconfigured as it does not reflect *a*'s expectation. Instead of denying *d*, she denied herself access to *z*. It contradicts requirements G2 (Allow everything the owner wants to be allowed) and G4 (Two rules within the same rule set must not conflict).
- The rules were generated by taking specific family members into account, instead of more general attributes, such as age. The use of attributes for generating small and understandable rule sets is recommended and is one of the reasons why ABAC is better suited for smart products, as mentioned in Section 2.4. Therefore, there is a contradiction with G5 (Minimize the number of rule set elements).

The smart products analyze the manually generated rule set, considering the usability and security constraints presented in Chapter 4, and

produce new rule sets that are free of conflicts. In our example, the smart products present to the user *a* two automatically generated rule set alternatives, as seen in Table 7.2 and Table 7.3.

Table 7.2.: Interactive Rule Learning Rule Set 2

1:	If (age > 40)	→	full access
2:	If (family & age > 16)	→	remote and local access
3:	If (age > 9)	→	local access

Table 7.3.: Interactive Rule Learning Rule Set 3

1:	If (parents)	→	full access
2:	If (family & age > 16)	→	remote and local access
3:	If (age > 16)	→	local access

It is up to *a* to decide which rule set suits her needs best. Both rule sets look much better and more concise than the manually generated rule set. However, the first rule of Rule Set 2 is way too general (a violation of requirement G1), since it gives full access rights for everyone above 40, including potential guests. The last rule of Rule Set 2 is also not to her liking, since *a* would not trust a 9-year old to operate kitchen appliances (but she would trust a 12-year old). Thus, *a* picks Rule Set 3, but manually changes rules 2 and 3 to better fit her expectations. The modified rule set, Rule Set 4, can be seen in Table 7.4.

Table 7.4.: Interactive Rule Learning Rule Set 4

1:	If (parents)	→	full access
2:	If (family & age > 12)	→	remote and local access
3:	If (age > 12)	→	local access

A comparison between the manually generated Rule Set 1 and the interactive generated Rule Set 4 demonstrates a great improvement of the latter regarding the usability and security requirements presented

in Chapter 4. Rule Set 4 addresses the security requirements G1 and G2 since the rules are specific and meaningful. Usability requirements G3, G4, G5, and G6 are also fulfilled since there are no redundant rules, and the rules are consistent, understandable, meaningful, manageable and express the owner's security expectations with a minimal amount of rules.

7.5. Conclusion

In this dissertation security and usability metrics are introduced that quantify how usable access control rule sets are (see definition of usable access control rule sets in Section 1).

A possible integration of usable access control into smart products is shown. Based on analysis of the different AC mechanisms, we propose the combination of a blacklist, an attribute based access control (ABAC) approach, and a cooperative intrusion detection system (IDS) further combined with Interactive Rule Learning in the future to satisfy current and future needs for smart products usability. A design description based on FMC diagrams showed the integration of the proposed security solution into the SmartProducts framework. The Access Handler and Security Administration design components directly correlate to the aforementioned principles of ABAC and Interactive Rule Learning. Use cases were provided for both components to demonstrate the dynamic structure of the smart products security design.

After performing an initial analysis of security objectives, gaps regarding the usability of access control rule set configuration were identified. Informal requirements for usable access control rule sets were refined out of a pilot study and related work.

These informal requirements were then formalized. A minimal set of basic formal building blocks was obtained as a set of six formal definitions for security and usability properties of access control rule sets. They provide tangible and simple sets, metrics, and optimization crite-

ria that reflect the characteristics and the errors in access control rule sets. The provided metrics were validated by user studies that resulted in statistically significant evidence supporting the hypotheses:

Hypothesis H1: The sets and metrics help users produce better rule sets.

Hypothesis H2: The usability scores computed using the metrics in Chapter 5 correspond to rankings obtained from IT support professionals when evaluating the translation of policies into access control rule sets (related to *G1* and *G2*).

Hypothesis H3: The usability scores computed using the metrics in Chapter 5 correspond to rankings obtained from IT support professionals when evaluating the understandability and manageability of access control rule sets (related to *G3*, *G4* and *G5*).

In conclusion, our approach offers a uniform and scientific method for comparing different rule sets. Moreover, our metrics can be used as optimization criteria to generate usable access control rule sets and to improve their manageability. Furthermore, a formalization is the first step towards the implementation of tools for automatically measuring and comparing different rule sets.

Future and ongoing work aims to demonstrate that the implementation of the results presented in this dissertation can be used to automatically improve rule sets. The objective of such work is to design tools that can be integrated into the everyday working environment to actively help users produce usable access control rule sets and to make Interactive Rule Learning possible.

A. Appendix

A.1. User Study Rule Sets

In this section the rule sets generated in User Study 1 are shown. The rule sets are ordered by the total cost of each rule set. The best rule set (which has the lowest cost) is at the top, the worst rule set is at the end. These are the rule sets that were sent in random order to the experts for User Study 2.

Table A.1.: User Study - Rule Set 1

#	Path	Attributes	Decision
1	C:\a\	A5	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\ a\ a\	A4 A5	ALLOW
5	C:\c\b\	A3	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\b\	A3 A4	ALLOW

Table A.2.: User Study - Rule Set 2

#	Path	Attributes	Decision
1	C:\a\	A5	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\a\b\	A6 A5 A7	ALLOW
5	C:\c\b\	A7 A3	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\c\	A8	ALLOW
8	C:\c\c\b\	A3 A4	ALLOW
9	C:\c\a\a\	A4 A5 A7	ALLOW

Table A.3.: User Study - Rule Set 3

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\a\	A6	ALLOW
5	C:\c\b\	A3	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\c\	A8	ALLOW
8	C:\c\b\	A6	ALLOW
9	C:\c\c\b\	A3 A4	ALLOW
10	C:\c\a\a\	A4 A5	ALLOW

Table A.4.: User Study - Rule Set 4

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\a\a\	A4 A5	ALLOW
5	C:\c\b\	A3	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\b\	A4 A3	ALLOW
8	C:\c\b\	A6	ALLOW

Table A.5.: User Study - Rule Set 5

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\a\a\	A4 A5	ALLOW
5	C:\c\b\	A3	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\b\	A4 A3	ALLOW
8	C:\c\b\	A6	ALLOW

Table A.6.: User Study - Rule Set 6

#	Path	Attributes	Decision
1	C:\a\	A7	ALLOW
2	C:\a\	A5	DENY
3	C:\b\	A7	ALLOW
4	C:\c\a\	A4 A5	ALLOW
5	C:\c\a\a\	A6	ALLOW
6	C:\c\a\b\	A6	ALLOW
7	C:\c\a\c\	A6	ALLOW
8	C:\c\b\	A3	ALLOW
9	C:\c\b\a\	A6	ALLOW
10	C:\c\b\b\	A6	ALLOW
11	C:\c\c\	A8	ALLOW
12	C:\c\c\c\	A8	ALLOW
13	C:\c\a\b\	A4	DENY
14	C:\c\a\c\	A4	DENY
15	C:\c\b\	A6	ALLOW
16	C:\c\c\	A6	ALLOW
17	C:\c\c\b\	A3 A4	ALLOW
18	C:\c\c\b\	A3 A4 A5	DENY

Table A.7.: User Study - Rule Set 7

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\ a\	A4 A5	ALLOW
5	C:\c\b\	A3	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\b	A5 A6	ALLOW
8	C:\c\ a\b\	A4 A5	DENY
9	C:\c\b\ a\	A8	DENY
10	C:\c\b\ a\	A4 A5	DENY
11	C:\c\c\b\	A3 A4	ALLOW

Table A.8.: User Study - Rule Set 8

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\ a\	A4 A5	ALLOW
5	C:\c\b\	A7	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\c\	A8	ALLOW
8	C:\c\b\ a\	A8	DENY
9	C:\c\b\ a\	A4 A5	DENY
10	C:\c\ a\b\	A4 A5	DENY
11	C:\c\ a\c\	A4 A5	DENY
12	C:\c\b\b\	A8	DENY
13	C:\c\b\b\	A4 A5	DENY
14	C:\c\b\c\	A8	DENY
15	C:\c\b\c\	A4 A5	DENY
16	C:\c\c\b\	A3 A4	ALLOW

Table A.9.: User Study - Rule Set 9

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\	A4 A5	ALLOW
5	C:\c\b\	A7	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\c\	A8	ALLOW
8	C:\c\	A4 A5	DENY
9	C:\c\b\	A8	DENY
10	C:\c\b\b\	A8	DENY
11	C:\c\b\c\	A8	DENY
12	C:\c\	A4 A5	DENY
13	C:\c\b\c\	A4 A5	DENY
14	C:\c\c\b\	A7	ALLOW
15	C:\c\b\	A3 A4 A5	ALLOW
16	C:\c\b\	A4 A5	DENY
17	C:\c\b\b\	A4 A5	DENY
18	C:\c\b\c\	A3	ALLOW
19	C:\c\b\c\	A6	ALLOW

Table A.10.: User Study - Rule Set 10

#	Path	Attributes	Decision
1	C:\a\	A3 A4 A7	ALLOW
2	C:\b\	A1	ALLOW
3	C:\c\a\	A4 A5	ALLOW
4	C:\c\b\	A7	ALLOW
5	C:\c\c\	A8	ALLOW
6	C:\c\c\c\	A8	ALLOW
7	C:\b\	A2	ALLOW
8	C:\b\	A3	ALLOW
9	C:\b\	A4	ALLOW
10	C:\b\	A5	ALLOW
11	C:\b\	A6	ALLOW
12	C:\b\	A7	ALLOW
13	C:\b\	A8	ALLOW
14	C:\c\c\b\	A3 A4 A7	ALLOW
15	C:\c\b\c\	A5 A7 A8	DENY
16	C:\c\b\c\	A4 A5 A7	DENY
17	C:\c\c\b\	A3 A4 A5 A7	DENY
18	C:\c\c\b\	A4 A5 A7	DENY
19	C:\	A6	ALLOW
20	C:\a\	A5 A5 A7	DENY
21	C:\c\c\c\	A3 A4 A5 A7	DENY
22	C:\c\c\c\	A4 A5 A7	DENY
23	C:\c\c\b\	A4 A5 A7	DENY
24	C:\c\c\b\	A5 A7 A8	DENY
25	C:\c\c\c\	A5 A7 A8	DENY
26	C:\c\c\c\	A4 A5 A7	DENY
27	C:\c\c\c\	A3 A4 A5 A7	ALLOW

Table A.11.: User Study - Rule Set 11

#	Path	Attributes	Decision
1	C:\a\	A3 A4	ALLOW
2	C:\a\	A5	DENY
3	C:\b\	A7	ALLOW
4	C:\c\	A6	ALLOW
5	C:\c\a\a\	A4 A5	ALLOW
6	C:\c\b\	A7	ALLOW
7	C:\c\b\	A8	DENY
8	C:\c\c\	A8	ALLOW
9	C:\c\c\b\	A3 A4	ALLOW

Table A.12.: User Study - Rule Set 12

#	Path	Attributes	Decision
1	C:\a\	A5	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\a\	A4 A5	ALLOW
5	C:\c\b\	A3	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\b\	A5	DENY
8	C:\c\c\b\	A3 A4	ALLOW

Table A.13.: User Study - Rule Set 13

#	Path	Attributes	Decision
1	C:\a\	A7	DENY
2	C:\c\a\a\	A4 A5 A7	ALLOW
3	C:\b\	A7	ALLOW
4	C:\a\b\	A6	ALLOW
5	C:\a\c\	A6	ALLOW
6	C:\c\b\a\	A7	DENY
7	C:\c\b\b\	A7	ALLOW
8	C:\c\b\c\	A7	ALLOW
9	C:\c\c\a\	A8	ALLOW
10	C:\c\c\b\	A3 A4 A7	ALLOW
11	C:\c\	A6	ALLOW
12	C:\b\	A6	ALLOW
13	C:\c\c\	A8	ALLOW
14	C:\c\b\	A8	DENY
15	C:\c\b\	A3	ALLOW
16	C:\c\b\	A6	ALLOW

Table A.14.: User Study - Rule Set 14

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\a\	A4 A5	ALLOW
5	C:\c\b\	A3	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\	A8 A7	ALLOW

Table A.15.: User Study - Rule Set 15

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\a\	A4 A5	DENY
5	C:\c\b\	A7	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\c\	A8	ALLOW
8	C:\c\b\a\	A8	DENY

Table A.16.: User Study - Rule Set 16

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\a\a\	A4 A5	ALLOW
5	C:\c\a\a\	A5 A6 A7	ALLOW
6	C:\c\b\	A7	ALLOW
7	C:\c\c\	A8	ALLOW
8	C:\c\c\c\	A8	ALLOW

Table A.17.: User Study - Rule Set 17

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\a\	A4 A5	ALLOW
5	C:\c\b\	A3	ALLOW
6	C:\c\b\	A6	ALLOW
7	C:\c\b\	A8	ALLOW
8	C:\c\c\b\	A3 A4	ALLOW
9	C:\c\c\b\	A6	ALLOW
10	C:\c\c\b\	A8	ALLOW
11	C:\c\c\c\	A8	ALLOW
12	C:\c\c\a\	A8	ALLOW

Table A.18.: User Study - Rule Set 18

#	Path	Attributes	Decision
1	C:\a\	A6	DENY
2	C:	A6	ALLOW
3	C:\b\	A7	ALLOW
4	C:\c\a\	A4 A5	ALLOW
5	C:\c\b\	A7	ALLOW
6	C:\c\c\	A8	ALLOW
7	C:\c\c\c\	A8	ALLOW
8	C:\c\b	A8	DENY

Bibliography

Proc. of the 29th Int. Conference on Human Factors in Computing Systems (CHI 2011), 2011. ACM. ISBN 978-1-4503-0228-9.

D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. In *Web Semantics: Science, Services and Agents on the World Wide Web 5.2*, pages 58–71, 2007.

L. Bauer, L. F. Cranor, R. W. Reeder, M. K. Reiter, and K. Vaniea. Real life challenges in access-control management. In *Proc. CHI 2009*, pages 899–908. ACM, 2009. ISBN 978-1-60558-246-7.

M. Beckerle. *Interaktives Regellernen*. Master thesis, Technische Universität Darmstadt, 2009a.

M. Beckerle. Towards Smart Security for Smart Products. In *AmI-Blocks'09: 3rd European Workshop on Smart Products*, 2009b.

M. Beckerle, editor. *Section 5.15, and 5.16, SmartProducts Deliverable D1.2.1: Initial Concepts for Smart Products [Restricted]*. July 2009c.

M. Beckerle, editor. *Section 5.5, and 6.7, SmartProducts Deliverable D4.1.1: Requirements Analysis for Storing, Distributing, and Maintaining Proactive Knowledge Securely [Restricted]*. July 2009d.

M. Beckerle, editor. *SmartProducts Deliverable D4.2.1: Initial Concept for Security and Privacy of Proactive Knowledge*. February 2010a. Retrieved: October 2013, from http://www.smartproducts-project.eu/media/stories/smartproducts/publications/SmartProducts_D4.1.2_D4.2.1_Final.pdf.

- M. Beckerle, editor. *Section 3.2, and 4.4 - 4.6, SmartProducts Deliverable D4.3.1: Specification of Services to Manage Proactive Knowledge [Confidential]*. February 2010b.
- M. Beckerle, editor. *Section 4, SmartProducts Deliverable D4.3.2: Initial Implementation of the Required Set of Services to Manage Proactive Knowledge [Restricted]*. August 2010c.
- M. Beckerle, editor. *Section 4, SmartProducts Deliverable D4.4.1: Evaluation Report for Initial Implementation [Confidential]*. November 2010d.
- M. Beckerle, editor. *SmartProducts Public Deliverable D4.2.2: Final Concept for Security and Privacy of Proactive Knowledge*. Feb 2011a. Retrieved: October 2013, from http://www.smartproducts-project.eu/media/stories/smartproducts/publications/SmartProducts_D4.2.2_Final.pdf.
- M. Beckerle, editor. *Section 2.2, and 6, SmartProducts Public Deliverable D4.3.3: Final Implementation of the Required Set of Services to Manage Proactive Knowledge [Confidential]*. November 2011b.
- M. Beckerle, editor. *Section 4, SmartProducts Public Deliverable D4.4.2: Evaluation Report for Final Implementation [Confidential]*. February 2012.
- M. Beckerle and L. A. Martucci. Formal definitions for usable access control rule sets from goals to metrics. In *Proceedings of the Ninth Symposium on Usable Privacy and Security, SOUPS '13*, pages 2:1–2:11, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2319-2. doi: 10.1145/2501604.2501606. URL <http://doi.acm.org/10.1145/2501604.2501606>.
- M. Beckerle, L. A. Martucci, and S. Ries. Interactive access rule learning: Generating adapted access rule sets. In *ADAPTIVE 2010 : The Second International Conference on Adaptive and Self-Adaptive Systems and Applications*, ComputationWorld 2010, pages 104–110. IARIA, Nov 2010. ISBN 978-1-61208-001-7.

- M. Beckerle, L. A. Martucci, S. Ries, and M. Mühlhäuser. Interactive rule learning for access control: Concepts and design. *International Journal on Advances in Intelligent Systems*, 4:234–244, 2011.
- D. E. Bell and L. J. L. Padula. Secure computer system: Unified exposition and Multics interpretation. *MTR-2997*, 1976.
- P. Bimmel, U. Rampillon, and H. Meese. *Lernerautonomie und Lernstrategien*. Langenscheidt, 2000.
- S. L. Brand. DoD 5200.28-STD Department of Defense Trusted Computer System Evaluation Criteria (Orange Book). *National Computer Security Center*, 1985.
- A. D. Brucker and H. Petritsch. Extending access control models with break-glass. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 197–206. ACM, 2009.
- J. G. Carbonell, R. S. Michalski, and T. M. Mitchell. *An overview of machine learning*. Tioga Publishing Company, Palo Alto, 1983.
- L. Cranor and S. Garfinkel. *Security and Usability*. O’Reilly Media, Inc., 2005.
- J. Daemen and V. Rijmen. AES proposal: Rijndael. 1999.
- H. Debar, M. Dacier, and A. Wespi. Towards a taxonomy of intrusion-detection systems. *Comput. Networks*, 31(8):805–822, 1999.
- C. Eckert. *IT-Sicherheit - Konzepte, Verfahren, Protokolle*. Oldenbourg, 2009. ISBN 978-3-486-58999-3.
- S. Egelman, A. Oates, and S. Krishnamurthi. Oops, I did it again: mitigating repeated access control errors on Facebook. In *Proc. CHI 2011 CHI* [2011], pages 2295–2304. ISBN 978-1-4503-0228-9.
- European Commission. Proposal for a regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation), 2012. Retrieved: October 2013, from http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf.

- D. F. Ferraiolo and D. R. Kuhn. Role-based access controls. In *Proc. of the 15th National Computer Security Conference*, pages 554–563, 1992.
- J. Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, 1999.
- T. Grandison and M. Sloman. A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 3:2–16, 2000.
- S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall, 3 edition, 2008.
- A. Herzog and N. Shahmehri. *New Approaches for Security, Privacy and Trust in Complex Environments*, pages 37–48. Springer Boston, 232 edition, 2007.
- G. E. Hinton and T. J. Sejnowski. *Unsupervised learning: foundations of neural computation*. The MIT press, 1999.
- G. Iachello and J. Hong. End-user privacy in human-computer interaction. In *Foundations and Trends in Human-Computer Interaction 1.1*, pages 1–137, 2007.
- X. Jin, R. Krishnan, and R. Sandhu. A unified attribute-based access control model covering dac, mac and rbac. In *Data and Applications Security and Privacy XXVI*, pages 41–55. Springer, 2012.
- Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 9(1):3–12, 2005.
- O. Kasten, M. Miche, D. Schreiber, M. Hartmann, A. Hadjakos, P. Hugeus, V. Uren, A.-S. Dadzie, J. Kantorovitch, E. Vildjiounaite, N. Ilkka, J. Mascolo, S. Luitjens, and A. Nikolov, editors. *Smart-Products Deliverable D12.1.3: Rolling Report on Use Cases and Trials*. February 2012. Retrieved: October 2013, from http://www.smartproducts-project.eu/media/stories/smartproducts/publications/SmartProducts_D12.1.3_Final.pdf.

- F. Keller and S. Wendt. Fmc: An approach towards architecture-centric system development. In *ECBS*, pages 173–182. IEEE Computer Society, 2003. ISBN 0-7695-1917-2.
- H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-hashing for message authentication, 1997.
- L. A. Martucci. *Identity and Anonymity in Ad Hoc Networks*. PhD thesis, Karlstad University, Jun 2009.
- L. A. Martucci, C. Andersson, and S. Fischer-Hübner. Chameleon and the Identity-Anonymity Paradox: Anonymity in Mobile Ad Hoc Networks. In *Proceedings of the 1st International Workshop on Security (IWSEC 2006)*, pages 123–134. Information Processing Society of Japan (IPSJ), 23–24 Oct 2006. ISBN 4-915256-59-6 C3040.
- L. A. Martucci, S. Ries, and M. Mühlhäuser. Sybil-Free Pseudonyms, Privacy and Trust: Identity Management in the Internet of Services. *Journal of Information Processing*, 19:317–331, Jul 2011. ISSN 1882-6652.
- M. L. Mazurek, J. P. Arsenault, J. Bresee, N. Gupta, I. Ion, C. Johns, D. Lee, Y. Liang, J. Olsen, B. Salmon, R. Shay, K. Vaniea, L. Bauer, L. F. Cranor, G. R. Ganger, and M. K. Reiter. Access control for home data sharing: evaluating social acceptability. In *Proc. CHI 2010*, pages 645–654. ACM, 2010. ISBN 978-1-60558-929-9.
- M. L. Mazurek, P. F. Klemperer, R. Shay, H. Takabi, L. Bauer, and L. F. Cranor. Exploring reactive access control. In *Proc. CHI 2011 CHI [2011]*, pages 2085–2094. ISBN 978-1-4503-0228-9.
- M. Mühlhäuser. Smart products: An introduction. In *Constructing Ambient Intelligence, AmI 2007 Workshops, Darmstadt, Germany*, pages 158–164. Springer, Heidelberg, Germany, 2008. Retrieved: October 2013, from <http://atlas.tk.informatik.tu-darmstadt.de/Publications/2008/Intro-SmartProductsWorkshop-AMI07.pdf>.
- R. S. Michalski. *Understanding the nature of learning: Issues and research directions*. Morgan Kaufmann, San Mateo, Calif, 1986.

- T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.
- R. Neisse, M. Wegdam, M. van Sinderen, and G. Lenzini. Trust management model and architecture for context-aware service platforms. In *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, Lecture Notes in Computer Science, pages 1803–1820. Springer Berlin Heidelberg, 2007.
- I. Ray, M. Kumar, and L. Yu. *LRBAC: A location-aware role-based access control model*. Springer, 2006.
- R. W. Reeder, L. Bauer, L. F. Cranor, M. K. Reiter, K. Bacon, K. How, and H. Strong. Expandable grids for visualizing and authoring computer security policies. In *Proc. CHI 2008*, pages 1473–1482. ACM, 2008. ISBN 978-1-60558-011-1.
- R. W. Reeder, L. Bauer, L. F. Cranor, M. K. Reiter, and K. Vaniea. More than skin deep: measuring effects of the underlying model on access-control system usability. In *Proc. CHI 2011* [CHI \[2011\]](#), pages 2065–2074. ISBN 978-1-4503-0228-9.
- M. Reiter and A. Rubin. Crowds: Anonymity for Web Transactions. In *DIMACS Technical report*, pages 97–115, 1997.
- M. Riedmiller. Advanced supervised learning in multi-layer perceptrons-from backpropagation to adaptive learning algorithms. *Computer Standards and Interfaces*, 16:265–278, 1994.
- R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- P. Samarati and S. D. C. di Vimercati. Access control: Policies, models, and mechanisms. In *Foundations of Security Analysis and Design, Tutorial Lectures (FOSAD 2000)*, volume 2171 of *Lecture Notes in Computer Science*, pages 137–196. Springer, 2000. ISBN 3-540-42896-8.

- R. S. Sandhu and P. Samarati. Access control: Principles and practice. *Communications Magazine, IEEE*, 32:40–48, Sep 1994. ISSN 0163-6804.
- B. Schoelkopf, C. Burges, and A. Smola. *Introduction to support vector learning*. MIT Press Cambridge, MA, USA, 1999.
- D. Schreiber, S. Z. Ali, M. Hartmann, I. Delchev, and M. Hillukkala, editors. *SmartProducts Deliverable D6.2.2: Final Architecture and Specification of Platform Core Services*. January 2011. Retrieved: October 2013, from http://www.smartproducts-project.eu/media/stories/smartproducts/publications/SmartProducts_D6.2.2_Final.pdf.
- H. A. Simon. *Why should machines learn?* Tioga Publishing Company, Palo Alto, 1983.
- SmartProducts. Smartproducts - proactive knowledge for smart products. Retrieved: October 2013, from <http://www.smartproducts-project.eu/index.php/vision>.
- D. K. Smetters and N. Good. How users use access control. In *Proc. SOUPS 2009*, ACM International Conference Proceeding Series. ACM, 2009. ISBN 978-1-60558-736-3.
- F. Stajano. *Security for ubiquitous computing*. John Wiley and Sons, 2002.
- Symantec. 2011 cost of data breach study, 2011. Retrieved: October 2013, from <http://www.symantec.com/content/en/us/about/media/pdfs/b-ponemon-2011-cost-of-data-breach-us-en-us.pdf>.
- P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy (S&P 1997)*, pages 44–54. IEEE Computer Society, 4–7 May 1997. ISBN 0-8186-7828-3.
- M. Weiser. The computer for the 21st century. *Scientific american*, 265 (3):94–104, 1991.

- A. Westin. *Privacy and Freedom*. Atheneum, first edition edition, 1967.
- E. Yair and A. Gersho. The boltzmann perceptron network: A soft classifier. *Neural Networks*, 3:203–221, 1990.
- E. Yuan and J. Tong. Attributed based access control (ABAC) for Web services. In *Proc. ICWS 2005*, pages 561–569, 2005a.
- E. Yuan and J. Tong. Attributed based access control (ABAC) for Web services. In *IEEE International Conference on Web Services ICWS 2005. Proceedings*, 2005b.
- L. Zhou, V. Varadharajan, and M. Hitchens. Cryptographic role-based access control for secure cloud data storage systems. In *Security, Privacy and Trust in Cloud Systems*, pages 313–344. Springer, 2014.

Affirmation / Ehrenwörtliche Erklärung

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades “Dr. rer. nat.” mit dem Titel “Usable Access Control” selbstständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Biblis, Oktober 2013

Wissenschaftlicher Werdegang des Verfassers

04/2002 – 04/2009 Studium der Physik an der Technischen Universität Darmstadt, Darmstadt

10/2002 – 04/2009 Studium der Informatik an der Technischen Universität Darmstadt, Darmstadt

10/2006 – 10/2008 Studium der Psychologie an der Technischen Universität Darmstadt, Darmstadt

10/2008 – 04/2009 Diplomarbeit am Lehrstuhl Knowledge Engineering der Technischen Universität Darmstadt, Darmstadt “Interaktives Regellernen”

04/2009 – 12/2013 Doktorand am Lehrstuhl für “Telekooperation” an der Technischen Universität Darmstadt, Darmstadt, Tätigkeit als Wissenschaftlicher Mitarbeiter und als Stipendiat, partielle Zugehörigkeit zum Center for Advanced Security Research Darmstadt - CASED.